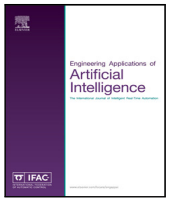
Engineering Applications of Artificial Intelligence 110 (2022) 104743

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Survey paper

A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects

Absalom E. Ezugwu [a,*], Abiodun M. Ikotun [a], Olaide O. Oyelade [a], Laith Abualigah [b,c,**], Jeffery O. Agushaka [a], Christopher I. Eke [d], Andronicus A. Akinyelu [e]

[a] School of Computer Science, University of KwaZulu-Natal, Pietermaritzburg Campus, Pietermaritzburg 3201, South Africa
[b] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[c] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, 11800, Malaysia
[d] Department of Computer Science, Faculty of Natural Science, Federal University of Lafia, Lafia, Nigeria
[e] Department of Computer Science and Informatics, University of the Free State, 9301 Bloemfontein, South Africa

ARTICLE INFO

Keywords:
Clustering
Clustering algorithms, partitioning
Data mining
Hierarchical clustering
Automatic clustering
K-Means
Optimization algorithms, Machine learning
Unsupervised learning
Supervised learning

ABSTRACT

Clustering is an essential tool in data mining research and applications. It is the subject of active research in many fields of study, such as computer science, data science, statistics, pattern recognition, artificial intelligence, and machine learning. Several clustering techniques have been proposed and implemented, and most of them successfully find excellent quality or optimal clustering results in the domains mentioned earlier. However, there has been a gradual shift in the choice of clustering methods among domain experts and practitioners alike, which is precipitated by the fact that most traditional clustering algorithms still depend on the number of clusters provided a priori. These conventional clustering algorithms cannot effectively handle real-world data clustering analysis problems where the number of clusters in data objects cannot be easily identified. Also, they cannot effectively manage problems where the optimal number of clusters for a high-dimensional dataset cannot be easily determined. Therefore, there is a need for improved, flexible, and efficient clustering techniques. Recently, a variety of efficient clustering algorithms have been proposed in the literature, and these algorithms produced good results when evaluated on real-world clustering problems. This study presents an up-to-date systematic and comprehensive review of traditional and state-of-the-art clustering techniques for different domains. This survey considers clustering from a more practical perspective. It shows the outstanding role of clustering in various disciplines, such as education, marketing, medicine, biology, and bioinformatics. It also discusses the application of clustering to different fields attracting intensive efforts among the scientific community, such as big data, artificial intelligence, and robotics. This survey paper will be beneficial for both practitioners and researchers. It will serve as a good reference point for researchers and practitioners to design improved and efficient state-of-the-art clustering algorithms.

1. Introduction

Clustering (an aspect of data mining) is considered an active method of grouping data into many collections or clusters according to the similarities of data points features and characteristics (Jain, 2010; Abualigah, 2019). Over the past years, dozens of data clustering techniques have been proposed and implemented to solve data clustering problems (Zhou et al., 2019; Abualigah et al., 2018a,b). In general, clustering analysis techniques can be divided into two main groups: hierarchical and partitional (Tan, 2018). Although methods in these two groups have proved to be very effective and efficient, they generally depend on providing prior knowledge or information of the exact number of clusters for each dataset to be clustered and analyzed (Chang et al., 2010). More so, when dealing with real-world datasets, it is normal not to expect or have any prior information regarding the number of naturally occurring groups in the data objects (Liu et al., 2011). Therefore, the concept of automatic data clustering algorithms is introduced to address this limitation. Automatic clustering algorithms refer to any clustering techniques used to automatically determine

* Correspondence to: School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, King Edward Road, Pietermaritzburg, KwaZulu-Natal 3201, South Africa.
** Corresponding author at: Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan.
E-mail addresses: Ezugwua@ukzn.ac.za (A.E. Ezugwu), 220078470@stu.ukzn.ac.za (A.M. Ikotun), oyeladeo@ukzn.ac.za (O.O. Oyelade), laythdyabat@aau.edu.jo (L. Abualigah), 218088307@stu.ukzn.ac.za (J.O. Agushaka), eke.ifeanyi@fulafia.edu.ng (C.I. Eke), AkinyeluAA@ufs.ac.za (A.A. Akinyelu).

https://doi.org/10.1016/j.engappai.2022.104743
Received 7 December 2021; Received in revised form 16 January 2022; Accepted 3 February 2022
Available online 23 February 2022

Abbreviations

The abbreviations used in this paper are given as listed below:

FCM	Fuzzy C Means
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
DENCLUE	DENsity-based CLUstEring
OPTIGRID	Optimal Grid
EM	Expectation Maximization
CLUSTREAM	Stream Clustering
CLUS TREE	Cluster Tree index structure for stream data
D-STREAM	Density Based Clustering for real time stream data
DENSTREAM	Density-Based Microclustering
DGCLUST	Distributed Grid Clustering
ODAC	Online Divisive–Agglomerative Clustering
LSEARCH	Linear Search
STREAMKM	K-means clustering algorithm for data streams
CURE	Clustering Using Representative
DIVFRP	Reference-point-based dissimilarity measure
IDPSO	Improved particle optimizer
HGCUDF	Hierarchical grid clustering using data field
SWIFT	Scalable Weighted Iterative Flow-clustering Technique
TR	Total Roughness
MMR	Min–Min-Roughness
MCA	Multiple Correspondence Analysis
MGR	Mean Gain Ratio
CF	Cluster Features
SS	Sum of Square
DTG	Delaunay triangulation graph
CLICK	CLuster Identification via Connectivity Kernels.
DBSCANS	Density Based Spatial Clustering of Applications with Noise
OPTICS	Ordering points to identify the clustering structure
MLE	Maximum Likelihood Estimation
SOM	Self-organizing map
DE	Differential Evolution
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
IWO	Invasive Weed Optimization
GKA	Genetic K-Means Algorithm
TWCV	Total within cluster variation
GGA	Genetically Guided Algorithm
IGKA	Incremental Genetic K-means Algorithm
ABC	Artificial Bee Colony Optimization Algorithm
SOS	Symbiotic Organisms Search
BEA	Bacterial Evolutionary Algorithm
IGKA	Incremental Genetic K-means Algorithm
SI	Swarm Intelligence
TS	Tabu Search
SA	Simulated Annealing
CF	Cluster features
HDENSTREAM	Density Based Clustering Algorithm for Heterogeneous Data Stream
C-DENSTREAM	A density-based clustering algorithm for data streams that includes domain information in the form of constraints
MUDI-STREAM	Multi Density data Stream
RDENSTREAM	Density Based Clustering Algorithm with outlier retrospect
HDDSTREAM	Density-based projected clustering algorithm for high dimensional data streams
SDSTREAM	Density-Based Data Streams Clustering over Sliding Windows
DDSTREAM	A Grid and Density-Based Clustering Algorithm for Processing Data Stream
DENGRIS	Density-Grid based Clustering Algorithm
D-STREAM	Stream Data Clustering Based on Grid Density and Attraction
PKS-STREAM	Data stream Clustering Algorithm based on grid density and index tree PKS tree
MR-STREAM	Density-Based Clustering of Data Streams at Multiple Resolutions
SWEM	An EM-Based Algorithm for Clustering Data Streams in Sliding Windows
GMM	Gaussian mixture model
GPU	Graphics Processing Unit
GMM	Gaussian mixture model
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
GDBSCAN	Generalized density based clustering algorithm
MRTA	Multi-Robot Task Allocation
ACD2PSO	A dynamic distributed particle swarm optimization

the number of clusters without having any prior information of the dataset features and attributes (Ezugwu, 2020a). Many automatic data clustering algorithms have been proposed in the literature, and several of them are nature-inspired. The current survey presents a systematic study of traditional and recently proposed clustering techniques applied in different fields.

Many surveys on clustering techniques exist in the literature (Xu and Wunsch, 2005; Xu and Tian, 2015; Benabdellah et al., 2019; Adil et al., 2014; Dafir et al., 2021; Saxena et al., 2017; Nagpal, 2013; Oyelade et al., 2016; Bindra and Mishra, 2017; Singh and Srivastava, 2020; Djouzi and Beghdad-Bey, 2019; Ezugwu, 2020a). Xu and Tian (2015) explained the basic elements involved in the clustering process and broadly categorized existing clustering algorithms into two major perspectives: the traditional and modern ones. Xu and Wunsch (2005) reviewed major clustering algorithms for datasets appearing in Statistics, Computer Science, and Machine learning. Benabdellah et al. (2019) categorized clustering algorithms using the three V's properties of Big Data: Volume, Variety, and Velocity. These three properties were used to explore the various categories of clustering algorithms. Adil et al. (2014) gave a concise survey of existing clustering algorithms and conducted extensive experiments to highlight the best-performing clustering algorithm for Big data analysis. Berkhin et al. (2001) reviewed clustering techniques in data mining, emphasizing object attribute type, large dataset scalability, handling high dimensional data, and finding irregularly shaped clusters.

Dafir et al. (2021)'s work was on parallel clustering algorithms, classifying and summarizing them. He discussed the framework for each kind of parallel clustering algorithm. Saxena et al. (2017) presented a

taxonomy of existing clustering algorithms, debating each algorithm's various measures of similarity and evaluation criteria. Nagpal (2013) carried out a comparative analysis of the different clustering algorithms concerning both the mixed and categorical datasets with the observation that no clustering algorithm can be adjudged as best for handling a large dataset of either the mixed or categorical dataset. Oyelade et al. (2016) examined various clustering algorithms and their suitability for gene expression data to discover and provide helpful knowledge that will guarantee stability and a high degree of accuracy in the area. Jain (2010) summarized well-known clustering methods with a discussion on critical issues and challenges in the design of clustering algorithms. Jain et al. (1999) discussed emerging techniques for non-numeric constraints and large sets of patterns. Ezugwu et al. (2020a) presented an in-depth and systematic review of nature-inspired metaheuristic algorithms used for automatic clustering analysis focusing on the metaheuristic algorithms that have been employed to solve clustering problems over the last three decades.

Obviously, from the literature, there has been a considerable growth of interdisciplinary interests and dynamics in the application of clustering analysis to different research domains indicating that without a doubt, much has been achieved regarding clustering with new emerging research directions in automatic clustering algorithms (Ezugwu et al., 2020a). However, despite the decades of reported research on clustering methods and algorithms, the existing literature is remarkably segmented. Moreover, applied researchers find it challenging to acquire systematic information on research progress and advancement on the subject (Ezugwu, 2020a). Therefore, there is the need for a comprehensive systematic survey of literature on both the traditional and recently proposed clustering techniques that have been applied in different fields. Hence, the following main research question for this study has been formulated as follow:

> "*What are the various state-of-the-art clustering methods and algorithms discussed in the literature, and in what research domains have they been applied?*".

Towards realizing the answer for the main research question, the following sub-research questions are formulated:

(a) What are the various *traditional and recently proposed clustering techniques and algorithms* in existence today?
(b) What research has been conducted using both the traditional and recently proposed clustering techniques to address *identified challenges of clustering*?
(c) In what domains have both the traditional and recently proposed clustering techniques been applied in solving clustering problems?
(d) How have various similarity measures been employed in traditional and *recently proposed clustering techniques*?
(e) What are the characteristic differences between the traditional and recently proposed clustering techniques that have been applied in different fields?
(f) What *are other challenges of clustering problems* yet to be explored by researchers in this research area?

This survey aims to provide an up-to-date comprehensive review of the different clustering techniques applied to many data mining-related fields. Retrospectively, we also highlight novel and most recent practical applications areas of clustering. This survey is intended to provide a convenient research path for new researchers, furnishing them with a comprehensive study on the various data clustering techniques and research progression over the years in clustering techniques. This survey will also help experts develop new algorithms for emerging challenges in the research area. The main contribution of this survey study is as follows:

- Provides an up-to-date comprehensive systematic review of the traditional and recently proposed clustering techniques that have been applied in different fields.
- Provides a concise presentation of concepts, architecture, and taxonomy of clustering algorithms.
- Presents a discussion on open recent research issues relating to clustering problems
- Defined possible future research trends and directions regarding the implementation and application of clustering algorithms in different research domains.

2. Methodology

This section presents the procedure used in selecting and reviewing the various clustering methods considered in this survey. In this comprehensive review process and methodology, the standard approach for systematic literature review was adopted and followed to ensure that the topic of interest is sufficiently covered and reduce bias on the review work. In this study, the literature review procedure proposed by (Weidt and Silva, 2016) was used in this paper. Moreover, the work (Thilakaratne et al., 2019) served as a guide. The search techniques, search keywords, databases, data sources, and the inclusion and exclusion criteria used in this survey are explained below.

2.1. Keywords

In order to obtain the relevant literature, different keywords were selected regarding the defined goal of this survey. Initially, several keywords were formulated but later streamlined to reflect the research objective. The keywords used in the extraction of articles include: "Clustering", "non-supervised classification", "Clustering Algorithms", "Clustering Methods", "Evolutionary Clustering Algorithms", "Nature Inspired Clustering Algorithms"; "Data Mining Algorithms"; "Clustering Applications Areas"; "Clustering in Data Mining". Each of the various clustering methods from the taxonomy was paired with the word "clustering" to search for articles solely on the technique. For example, "mode seeking clustering", "subspace clustering, etc. The various application areas discussed were also paired with the term "Clustering" to search for articles that reported clustering activities in the field and the clustering algorithm employed.

2.2. Searching the articles

Two different rounds of the search were performed. The first round of searches was carried out from 11th to 28th of November 2020, while the second round of questioning was conducted between the 7th to 18th of December 2020. More searches were conducted in late December 2020 through mid-January 2021. During the search, more related papers were extracted from the citations of the selected relevant articles.

2.3. Academic databases

The formulated keywords were used in the retrieval of the literature. Reputable peer-reviewed journals, conference proceedings, and edited books indexed in different academic databases were targeted in the study. Table 1 below contains the list of the academic databases targeted during the search. No specific interval period was stated at the beginning of the investigation. This enables proper capturing and in-depth study of existing traditional clustering methods, which served as a bedrock for the more recent ones. However, in selecting previous related reviews and clustering applications, the selected articles were streamlined to the articles published from 1999 to date. This is to capture updated works on unsupervised classification techniques for different domains.

Table 1
The list of academic databases with their corresponding links.

Academic databases	Web ink
ACM digital library	http://dl.acm.org/
DBLP	http://dblp.uni-trier.de/
IEEE Xplore	http://ieeexplore.ieee.org/
ISI Web of science	http://apps.webofknowledge.com/
ScienceDirect	http://www.sciencedirect.com/
Scopus	http://www.scopus.com/
SpringerLink	http://www.link.springer.com/
Wiley online library	http://onlinelibrary.wiley.com/

2.4. Article Inclusion/Exclusion criteria

There is the need to set up some inclusion and exclusion criteria to extract only the relevant articles for this review. Articles' titles, abstracts, conclusions, and complete content were studied to decide the articles included or excluded. The criteria used for the exclusion and inclusion are presented in Table 2.

2.5. Eligibility

All the criteria outlined in Table 2 were used to determine the eligibility of the selected articles. Duplicates were removed, and many articles were eliminated based on their title. Some articles were eliminated after a careful study of the abstract and conclusion, while some were eliminated based on the study of their full content. A total number of 347 papers were considered eligible and hence used for this review.

3. Comparison with existing survey works

This section presents and discusses the main difference between the already published review papers and this survey paper. Although there have been several attempts in the literature to comprehensively present a systematic review of different clustering techniques with their trending applications areas, this effort has become almost impossible because of the evolving nature of the study area and its relevance to different theoretical and practical fields of study relative to data mining and knowledge acquisition. For example, Jain et al. (1999) published a relatively comprehensive overview of pattern clustering methods from a statistical pattern recognition perspective to provide useful advice and references to fundamental concepts accessible to the broad research community of clustering practitioners. Ezugwu et al. (2020a) presented a corpus of nature-inspired algorithms covering different algorithms, including stochastic, evolutionary, physical, probabilistic, swarm, immune and neural algorithms often employed for automatic clustering tasks application. A summary of the previous clustering survey studies compared to the present work is presented in table form, as illustrated in Table 3.

Despite researchers' efforts to bring together a comprehensive list of both traditional state-of-the-art clustering methods and the new generation clustering algorithms, only a handful of these survey papers covered an up-to-date report on vast clustering methods available in the literature comprehensively. Moreover, other attempted survey works covered limited classification of clustering algorithms. In the past few decades, there has been exponential growth in the design and implementation of new clustering algorithms scattered in the literature. The previous works mainly focused on either problem-specific clustering approaches or presented survey works that do not include the most recently published state-of-the-art clustering algorithms. More so, none of the earlier works attempted taxonomical analysis of the comprehensive list of the clustering algorithms to show the impact of the algorithms in different domains despite the significance of clustering analysis techniques in the literature. Table 4 summarizes the previous survey works with their current citation impact extracted from Google scholar.

4. Taxonomy of clustering algorithms

Several clustering algorithms have been identified and broadly classified under two categories, namely: the Hierarchical Clustering Algorithm and the Partitional Clustering Algorithm (Xu and Wunsch, 2005; Xu and Tian, 2015; Benabdellah et al., 2019; Adil et al., 2014; Dafir et al., 2021; Saxena et al., 2017; Nagpal, 2013; Oyelade et al., 2016; Bindra and Mishra, 2017; Singh and Srivastava, 2020; Djouzi and Beghdad-Bey, 2019; Ezugwu, 2020a). A hierarchical clustering algorithm is further subdivided into agglomerative and divisive methods, while the Partitional has four subdivisions: Hard or Crisp clustering method, Fuzzy method, Mixture, and Square error. Under the Hard or Crisp, six major categories are identified: the Search-based method, the Graph-theoretic method, Density-based, Model-based, Sub-space, and Miscellaneous. The complete taxonomy for these clustering methods is shown in Fig. 1.

4.1. Hierarchical clustering algorithms

In Hierarchical Clustering algorithms, data objects are partitioned into levels in a hierarchical format (Btissam, 2015). Clusters are iteratively formed in a top-down or bottom-up approach to generate a dendrogram depicting the formulated clusters' hierarchical structure (Saxena et al., 2017). This clustering method allows exploring data on different levels of granularity (Bekhin, 2006). The bottom-up approach is termed the agglomerative method, while the top-down approach is the divisive method. In the agglomerative method, clusters are built up from single objects that are iteratively merged appropriately into larger clusters that form the hierarchy's various levels until the entire object forms a single cluster or the stopping criterion is met. The reverse is the case in a divisive method. The cluster containing all the objects is broken down iteratively appropriately until each object forms a single cluster or the stopping criterion is met. The merging or splitting is done based on the similarity or dissimilarity of the cluster elements. Fig. 2 illustrates a dendrogram representation for the hierarchical clustering method.

In hierarchy clustering, merging or splitting subsets of a point is done by generalizing the distance between individual points to the distance between subsets of point. This is determined using a proximity measure called the linkage metric. Three basic linkage metrics are used in hierarchical clustering: the single linkage, the average connection, and the complete linkage (Saxena et al., 2017; Olson, 1995; Jain et al., 1999; Murtagh, 1985). The hierarchical clustering algorithm uses $N \times N$ connectivity matrix form, where the linkage metrics used for the clustering are constructed. The construction of the similarity matrix is achieved by finding the similarity between each pair of data points. The linkage criterion is then calculated by finding the pairwise distance between the clusters. The similarity metric is used to determine the distance between the sets of the clusters. It is also used in determining the shape of the clusters.

Single-linkage clustering: The single-linkage clustering is also referred to as the nearest neighbor or minimum or connectedness method. It measures the nearest distance from any member of one cluster to any other cluster member. It measures the similarity between two clusters by measuring the closest distance between a single element pair. The single linkage clustering has a chaining effect with the tendency of producing elongated clusters (Punit, 2018).

Average linkage clustering: The average linkage clustering is also regarded as the minimum-variance linkage (Saxena et al., 2017; Jain and Dubes, 1988). It finds the mean or median of the distances among all the data points between clusters (Punit, 2018).

Complete linkage: The complete linkage, also known as the maximum or diameter or farthest neighbor method, determines the distance between two clusters by measuring the longest distance from any member of one

Table 2
The inclusion and exclusion criteria used.

Inclusion	Exclusion
The review focused on the various Clustering methods and algorithms.	Articles on general Data Mining techniques were not considered.
Articles that reviewed specific clustering methods and their variants.	Articles focusing on the review of the performance of a specific method in a specific domain were excluded.
Articles on the application of a specific method in a specific domain.	Articles that focused on the comparison of various methods in a specific domain were not considered
Only articles written in the English language were considered.	Articles written in other languages were excluded from the review
Only published articles from reputable peer-reviewed journals, conference proceedings and edited books were considered.	Keynote speeches, PowerPoint presentation slides were not considered.

Table 3
Summary of the previous related review on metaheuristic algorithms.

Reference	Publication date	Remark
Jain et al. (1999)	1999	Areas covered include pattern representation, similarity computation, grouping process, and cluster representations. Other areas include statistical, fuzzy, neural, evolutionary, and knowledge-based approaches for clustering. Application areas covered include image segmentation, object recognition, document retrieval and data mining.
Murtagh (1983)	1983	The study presented an in-depth survey of agglomerative hierarchical clustering algorithms and discussed efficient implementations in R and other software environments. Similarly, a review of grid-based clustering focusing on hierarchical density-based approaches was also presented.
Belkin et al. (2006)	2006	Covered a few classical and evolutionary clustering algorithms with their associated challenges to the identified practical application areas
José-García and Gómez-Flores (2016)	2016	A total of 65 automatic clustering approaches were reviewed based on single-solution, single-objective, and multiobjective metaheuristics, whose usage percentages are 3%, 69%, and 28%, respectively.
Saxena et al. (2017)	2017	Presented a comprehensive study on existing clustering methods and developments made at various times. The similarity and the evaluation criteria, which are the central components of clustering, are also presented in the paper.
Bandaru et al. (2017)	2017	The paper surveyed the different data mining methods that can be applied to extract knowledge about multi-objective optimization problems from the solutions generated during optimization.
Yang and Wang (2018)	2018	This paper serves as an introductory text and survey for multi-view clustering. Simultaneously, the study summarizes many multi-view clustering algorithms and provides a taxonomy according to the mechanisms and principles involved.
Hancer et al. (2020)	2020	The paper introduces a comprehensive survey on feature selection approaches for clustering, reflecting on some advantages/disadvantages of current approaches from different perspectives and identifying promising trends for future research.
Ezugwu et al. (2020a)	2020	Presented a systematic taxonomical overview and bibliometric analysis of trends and progress in nature-inspired metaheuristic clustering approaches from the early attempts in the 1990s until today's novel solutions
Bhattacharjee and Mitra (2021)	2021	The survey presents a comprehensive study of various Density-based clustering algorithms (DBCLAs) over last the two decades, along with their classification.
This study	**2021**	Presentation of a comprehensive survey, taxonomy, analysis of state-of-the-art clustering algorithms that have ever been identified in literature till today's novel development and their practical applications areas.

cluster to any member of the other cluster. The complete-linkage algorithm clusters are more compact and tightly bound than single-linkage clustering (Jain and Dubes, 1988).

In calculating the inter-cluster distances, the three proximity as mentioned earlier measures consider all the points of a pair of clusters. They are regarded as graph methods (Xu and Wunsch, 2005). Sibson (1973) implemented the single linkage hierarchical clustering algorithm to produce SLINK, Voorhees (1986) and Defays (1977) implemented Voorhees' method and CLINK, which are the implementations of average link clustering algorithm and complete link clustering algorithm, respectively. Using the central point for determining proximity measure, other geometrics methods were developed based on the same idea. These included the median linkage, centroid linkage, and the minimum variance linkage metrics (Berkhin, 2001; Murtagh, 1983; Day and Edelsbrunner, 1984). A distance-based proximity measure captures the inter-cluster closeness while the similarity measures capture the intra-cluster connectivity.

The hierarchical clustering algorithms can easily handle any similarity measure and flexible level of granularity (Bekhin, 2006, Punit, 2018). As a result of these, it applies to any attribute type. The hierarchical clustering methods are plagued with irreversible split or merge processes such that already formulated clusters cannot be revisited to reassign wrongly assigned objects. The application of hierarchical clustering in large-scale data sets is limited because it has high computational complexity. According to Saxena et al. (2017), most of the hierarchical clustering has at least a computational complexity of ($O(N^2)$). Bekhin (2006) reported that hierarchical clustering algorithms based on linkage metrics suffer from time complexity. Apart from these, there is also the problem of the vagueness of termination criteria (Bekhin, 2006) and lack of robustness due to its sensitivity to noise and outliers. It has also been reported that linkage metrics-based hierarchical clustering algorithms that use Euclidean distance tend to form spherical shapes (Bekhin, 2006). Fraley (988), in their work, observed that "A drawback of agglomerative methods is that those that are practical in terms of time efficiency require memory usage proportional to the square of the number of groups in the initial partition".

4.1.1. Agglomerative hierarchical clustering

In the agglomerative method, clusters are built up from single objects, which are iteratively merged appropriately into larger clusters that form the hierarchy's various levels until the entire object includes a single cluster or until the stopping criterion is met. The single cluster forms the root of the hierarchy. Two closest objects (based on the similarity measure used) are combined to form a cluster during the merging operation. It requires at most n iterations to complete the clustering operation since two clusters are merged during one iteration.

Ward's clustering method implements an agglomerative clustering algorithm that is not based on the linkage metric implemented by Ward in 1963 (Ward, 1963). It was based on K-Means' objective function with the merger decision dependent on its effect on the objective

Table 4
Summary of major survey literature on clustering algorithms.

Clustering methods	Study covered	Application area	Author and year	Impact as of 2021
Clustering algorithms with design concepts that are based on the following: graph theory, combinatorial search techniques, fuzzy set theory, neural networks, and kernels techniques	Survey of clustering algorithms for data sets appearing in statistics, computer science, and machine learning. Several tightly related topics, such as proximity measure and cluster validation, were also covered in the paper	Benchmark data sets, such as the traveling salesman problem and bioinformatics	Xu and Wunsch (2005)	6369
FCM, BIRCH, DENCLUE, OptiGrid, and EM	The survey provided a comprehensive study of the clustering algorithms proposed in the literature, namely Fuzzy-Cmeans (FCM), BIRCH algorithm, DENCLUE algorithm, Optimal Grid (OptiGrid), and Expectation-Maximization (EM)	Big data clustering covering	Adil et al. (2014)	785
Traditional and modern clustering algorithms	The survey covered at least 19 categories of the commonly used clustering algorithms, with high practical relevance and well-studied in the literature.	Generic application	Xu and Tian (2015)	680
Hierarchical and nonhierarchical methods	General discussion and presentation of clustering algorithms specifically targeted at information retrieval applications	Document collection	Rasmussen (1992)	**677**
Text clustering algorithms including Hierarchical methods	Provided a detailed survey of the problem of text clustering. The key challenges of the clustering problem, as it applies to the text domain were discussed as well.	Text clustering relative to social network and linked data	Aggarwal and Zhai (2012)	653
BIRCH, CluStream, ClusTree, D-Stream, DenStream, DGClust, ODAC, Scalable K-Means, Single-pass K-Means, Stream, Stream LSearch, StreamKM++, SWClustering	Presented a survey of data stream clustering algorithms, providing a thorough discussion of the main design components of state-of-the-art algorithms.	Network intrusion detection, sensor networks, and stock market analysis	Silva et al. (2013)	463
Evolutionary algorithms, physical algorithms, swarm intelligence, bio-inspired algorithms, and other nature inspired algorithms	Provided an up-to-date review of all major nature-inspired metaheuristic algorithms that were employed for partitional clustering	Character recognition, traveling salesman problem, blind channel equalizer design, human action classification, book clustering, texture segmentation, tourism market segmentation, analysis of gene expression patterns, electrocardiogram processing, the security assessment in power systems, manufacturing cell design, clustering of sensor nodes, identification of clusters for accurate analysis of seismic catalogs.	Nanda and Panda (2014)	374
Fuzzyc-means, Agglomerative hierarchical, K-Means, fuzzyc-Means, K-Medoids-based genetic clustering, Neural network clustering performed by a batch EM version of minimal free energy vector quantization	The paper provided surveys and summaries of most previous works investigating the clustering of time series data in various application domains.	Business and socio-economics, engineering, science, medicine, art and entertainment	Liao (2005)	2524
Evolutionary algorithms	This paper provides an up-to-date overview of evolutionary algorithms for clustering, including advanced topics such as multiobjective and ensemble-based evolutionary clustering. Similarly, the study also provides a taxonomy that highlights some important aspects in the context of evolutionary data clustering, namely, fixed or variable number of clusters, cluster-oriented or non-oriented operators, context-sensitive or context-insensitive operators, guided or unguided operators, binary, integer, or real encodings, centroid-based, medoid-based, label-based, tree-based, or graph-based representations, among others.	Image processing, computer security, and bioinformatics	Hruschka et al. (2009)	731

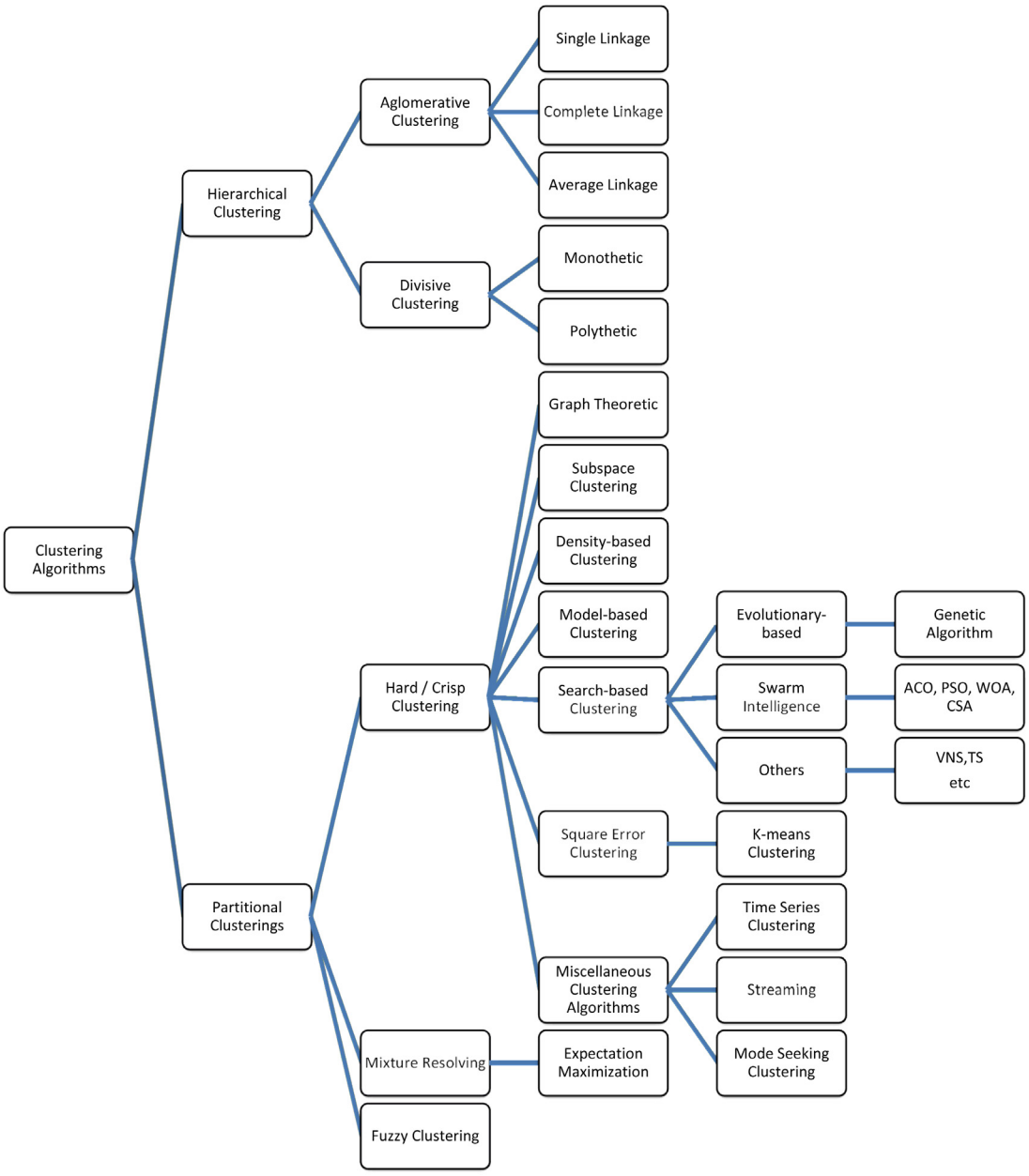


Fig. 1. Taxonomy of clustering algorithms.

function. Furthermore, it is considered a general agglomerative hierarchical clustering procedure. The criterion for choosing the pair of clusters to merge at each step is based on the optimal value of an objective function. This clustering method is most appropriate for quantitative variables and not binary variables. Gowda and Krishna (1978) developed a non-parametric hierarchical, agglomerative clustering algorithm based on the use of a conventional nearest neighbor to determine the mutual neighborhood value (MNV) and mutual nearest neighbors (MNN) of a sample point. Their simple, non-deterministic and non-iterative algorithm requires low storage and can discern non-spherical and spherical clusters. More so, their method was reported to have the ability to discern mutually homogeneous clusters and applications to a wide class of data of arbitrary shape, large size, and high dimensionality.

The Clustering Using Representative (CURE) implements agglomerative hierarchical clustering algorithms for large databases. It is more robust to outliers and can identify clusters of different sizes and shapes but with lesser cluster quality than BIRCH. It has a time complexity of $O\left(N^2 log N\right)$ and its performance is good on a 2-dimensional data set. CURE achieves scalability by using data sampling and partitioning; clusters with fine granularity are first constructed in partitions. Clusters are represented by a fixed number of points scattered around it. The distance between two clusters is generated by finding the minimum distance between the two clusters' representative points. The use of scattered representative points enables CURE to identify clusters of diverse sizes and shapes. The scattered representative points are shrunk to the cluster's geometric centroid as the clustering progresses based on the user-specified factor. The choice of the input parameters for CURE: the shrink factor, representative point number, sample size, and the number of partitions affect the clustering output. CURE was developed to work with a dataset with numerical attributes (Berkhin, 2001).

4.1.2. Divisive hierarchical clustering

Divisive hierarchical clustering is a reverse of the agglomerative clustering process that effectively divides every cluster into smaller chunks beginning with every object in a single cluster until the required

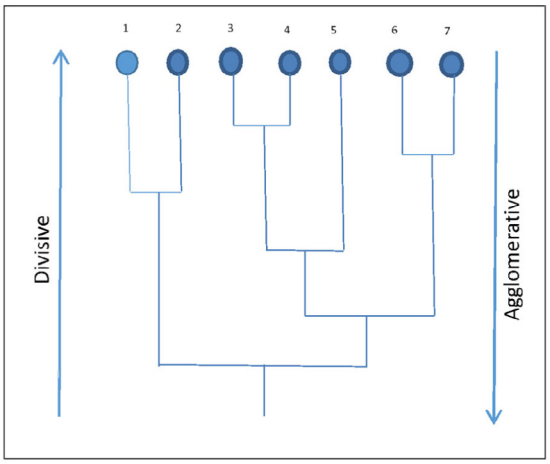


Fig. 2. A dendrogram representation for hierarchical clustering of data objects 1, 2, 3, 4, 5, 6, 7..

number of clusters is attained. By contrast with the agglomerative clustering method, the divisive approach uses the top-down method whereby the data objects are primarily considered a fused cluster that progressively divides pending when the cluster number is acquired (Boley, 1998; Savaresi et al., 2002; Chavent et al., 2007). The standard method of splitting a cluster into two subsets that contains one or more elements requires the consideration of every likely bipartition. Though it is normal to analyze all the likely bipartitions, whereby each cluster can be split into two sub-cluster, it is evident that the full enumeration process offers a universal optimum but very expensive in terms of computation cost. Various divisive clustering approaches that do not consider all bipartitions have been investigated. For instance, Karypis and Kumar (2000) proposed a bisecting K-Means divisive clustering method to attain more accurate results than the traditional K-Means or agglomerative method. In another study, Zhong et al. (2008) investigated a novel clustering method called "reference-point-based dissimilarity measure" (DIVFRP) by combining it with the divisive clustering method purpose of datasets partition. An improved particle optimizer (IDPSO) was proposed by Feng et al. (2010) to determine the closest optimal partition hyperplane for splitting designated clusters into two smaller chunks. The divisive hierarchical approach that uses this splitting method is both practical and efficient. Macnaughton-Smith et al. (1964) and Kaufman and Rousseeuw (2009) used an average dissimilarity between an object and a set of objects to investigate the iterative divisive procedure. However, a different approach that uses a dissimilarity matrix as an input relies on the optimization criteria that include the partition or bipartition (Guénoche et al., 1991; Wang et al., 1996).

Divisive clustering can be classified into two primary, monothetic, or polythetic methods. A divisive cluster is monothetic if combining a logical characteristic of each one involving one variable is essential and adequate for cluster membership (Sneath and Sokal, 1973). Monothetic divisive clusters are acquired by employing a single variable in each splitting by partitioning objects with specific object values from those without value. Monothetic is usually a variant of the "association analysis method" (Williams and Lambert, 1959) and is proposed for binary data. Various studies have applied monothetic clusters for problem-solving. For instance, Kim (2009) and Brito and Chavent (2012) employed a monothetic clustering approach on interval and histogram data. Similarly, Kim and Billard (2012) utilized the monothetic clustering method on multi-modal data. However, those monothetic approach decreases the number of computations required to identify an optimum bipartition, such that only $p(n-1)$ bipartition is needed for testing to determine the optimum bipartition instead of every $\left(2^{n-1}-1\right)$ likely bipartitions. The larger the number of objects, the possible bipartition number is further decreased by the monothetic method. Besides, they offer binary questions that facilitate the interpretation of clustering structures. On the other hand, the polythetic divisive clustering approach utilizes all variables concurrently via dissimilarity or distance values. It does not rely on single variable order but depends entirely on distance values, and the distance values reflect on all variable dissimilarity concurrently (Kim and Billard, 2011).

Various divisive clustering is polythetic such that all variables are considered concurrently in tasks to reveal the two instances' similarities (Kim et al., 2017). However, a large variable may lead to a scalability issue when there is a large variable. In this case, monothetic clustering that works on a single feature in a time cycle is the best option. Wang et al. (2014) proposed a modern divisive clustering algorithm termed 'Hierarchical grid clustering using data field' (HGCUDF). In this approach, hierarchical grids divide and conquer large datasets in their subset's hierarchy. However, the clustering regions limit the search scope, minimizing the data space for producing data fields. HGCUDF exhibits rapid execution and stability, which improves the clustering results on a large automated dataset. In another study, Naim et al. (2014) investigated a model-based clustering technique for high-dimension datasets. Its operation is divided into three stages: multi-modal splitting, iterative weighted sampling, and uni-modality preserving merging to measure the model-based clustering approach of large high-dimensional datasets. This method of clustering algorithm solves the problem of small datasets and the effective scaling of the large datasets when evaluated with synthetic datasets compared with the conventional methods. It is helpful in immune response jobs and can tremendously regulate rare populations.

Most of the clustering algorithms in the literature have focused on binary data. Recently, the clustering of categorical data has attracted most researchers in finding a solution to the definite data-clustering problem. Many researchers have proposed different divisive hierarchical clustering algorithms to combat the problem. For instance, Herawan et al. (2010) suggested 'Maximum Dependency of Attributes' (MDA) for divisive hierarchical clustering attributes selection. The maximum dependency of attributes is created by relying on attributes dependency in rough set theory, which measures the dataset's attributes dependency. Mazlack et al. (2000) investigated a bi-clustering method for choosing two-valued attributes by considering multi-valued attributes and a Total Roughness (TR) approach. They maintained that high total roughness attributes attain optimum performance and are suitable for cluster splitting. In another study, Parmar et al. (2007) developed a Min–Min-Roughness (MMR) metric to resolve the uncertainty in the categorical data clustering process. However, MMR signifies TR's reverse and does not yield clustering algorithms with comparative improvement in complexity or accuracy (Herawan and Deris, 2009; Herawan et al., 2010). Xiong et al. (2009) investigated a divisive method for categorical data based on "Multiple Correspondence Analysis" (MCA). Similarly, Qin et al. (2014) implemented information theory-based divisive clustering for categorical data by employing Mean Gain Ratio (MGR) to choose clustering attributes and select class equivalents on the cluster attribute using cluster entropy. Although divisive clustering is appealing based on computational time, partitioned clusters' quality is better than a divisive one.

Naim et al. (2014) proposed a model-based SWIFT (scalable weighted iterative Flow-Clustering Technique) for high-dimensional large datasets. The model consists of three stages, namely, multimodality splitting, iterative weighted sampling, and unimodality preserving merging for model-based clustering scaling on high-dimensional datasets are constructed to be effectively scalable to large datasets, offering a significant enhancement when compared with the current soft clustering approaches (Lo et al., 2008; Ge and Sealfon, 2012). These three major SWIFT stages are motivated by two main requirements: rare population identification and scalability to large datasets. In SWIFT, multimodality splitting and weighted iterative sampling identify rare populations. This algorithm is usually met for Flow Cytometry

(FC) and finding rare populations. The multimodality stage plays a vital role in identifying rare subpopulations. When evaluated with synthetic datasets, the algorithm solves small datasets and can effectively scale large datasets compared to conventional methods. SWIFT may also be employed to represent skewed clusters by LDA-based agglomerative merging, which decreases clusters numbers as it preserves the separate unimodal populations.

The interaction between the merging and multimodality splitting in many clusters uses a reasonable heuristic (cluster modality). It is more reasonable when compared with knee point in entropy plots formally employed (Lo et al., 2008; Finak et al., 2009). The algorithm advantageous immune response tasks and efficient scaling on large FC datasets. Moreover, the soft clustering method utilized in SWIFT is essential for understanding the overlapping clusters compared to the complex clusters approach like K-Means (Murphy, 1985) or spectral clusters (Zare et al., 2010). SWIFT has the power to control the tremendously rare populations. SWIFT is partially synonymous with flowPeaks (Ge and Sealfon, 2012) since both depend on unimodality criterion. Thus, flowPeaks focuses on the significant peaks without modality splitting and leans in missing tiny overlapping clusters. Consequently, one of the limitations of SWIFT is that it is restricted to a specific clustering task (Naim et al., 2014).

4.1.3. Some implementations for improving hierarchical clustering

The traditional algorithm has been enhanced over time to overcome the deficiencies of hierarchical algorithms. One improvement that takes hierarchical clustering limitation (due to large dataset) into consideration is the Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) clustering algorithm (Zhang et al., 1996). Birch employed the idea of Cluster Features (CF) which is presented as a triple containing the cluster objects total number n, the linear sum of attribute values of the cluster objects LS, and the sum of squares of the attribute's values of the cluster object SS, CF (n, LS, SS). The CF triple is a data structure that summarizes the information maintained about a cluster (Oyelade et al., 2016). CF triple is kept in a tree form, and only the tuples are kept in the main memory.

BIRCH is reported to be robust to outliers and can achieve O(N) computational complexity (Nagpal, 2013). The BIRCH algorithm consists of four phases, with phases 2 and 4 being optional (Oyelade et al., 2016). The scanning of the entire dataset and the construction of the CF tree is handled in phase 1. The clustering information stored in the CF Tree is done in such a way as to adequately reflect all the information in the dataset as much as possible while still accommodating the limitation imposed by the memory space with crowded data points grouped as fine sub-clusters. During the CF tree formation, outliers are treated as sparse data points and are removed from the dataset. The generation of the CF tree in phase 1 ensures no other input–output operation is required in the subsequent phases, thus reducing the computation time for the remaining steps. The clustering activity is also reduced to smaller sub-datasets of each sub-clusters in the entries of leaves of the CF tree because these are generated through incremental updating of the CF (Oyelade et al., 2016). The order of leaf entry of the initial tree construction produces a better data locality, enhancing the clustering output. BIRCH algorithm is credited with the ability to handle outliers, large datasets, and a good clustering output that is not affected by order of input data. The efficiency of BIRCH is, however, dependent on proper parameter settings. It also has the problem of biases towards non-spherical clusters due to diameter/radius in controlling the cluster boundary. Evaluation of BIRCH using both synthetic and real datasets showed that it returns a better result in computational time complexity, the robustness of the approach, and cluster quality.

COBWEB is another implementation of the hierarchical clustering algorithm for categorical data characterized with two major vital qualities: incremental learning and clusters' ability to be modeled or described intrinsically instead of being regarded as a collection of points. By processing one data point at a time, Cobweb dynamically builds the hierarchy of clusters instead of the merging or splitting approach of the agglomerative or divisive methods.

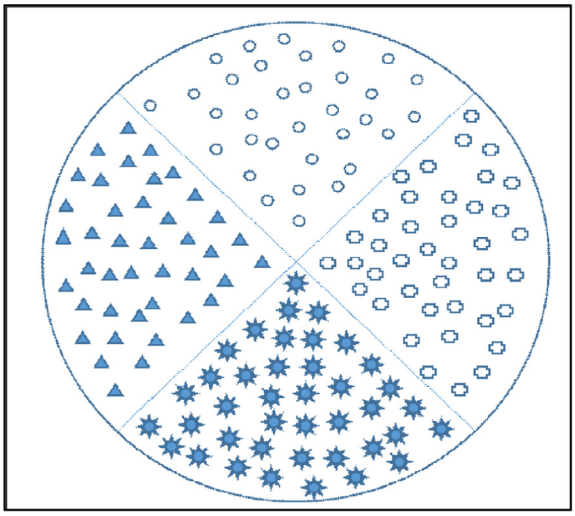


Fig. 3. A partition with $n = 154$ and $k = 4$.

4.2. Partitional clustering algorithm

In a partitional clustering algorithm, data is organized into a nested sequence of groups without any hierarchical structure (Jain and Dubes, 1988; Jain, 2010). Jain et al. (1999) stated that the partitioning method is suitable for handling clustering problems in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive. Their operation is based on generating data clusters to recover the existing natural groupings inherent in the dataset. Fig. 3 illustrates the clustering pattern representation of the partitional clustering method.

The dataset of n objects is iteratively partitioned into a predetermined k number of distinct subsets through the process of optimization of a criterion function (Ahmad and Dey, 2007). The squared-error criterion is the criterion function on which the most commonly used partitional clustering algorithm is based. The general objective is to find the partition for which a fixed number of clusters minimizes the square error. In this case, deviations of the patterns from the centroids are represented by the error with a view of the patterns as a collection of k numbered spherically shaped clusters. The target cost function ζ which can be minimized is given in Eq. (1):

$$\zeta = \sum_{i=1}^{n} \left\| d_i - C_j \right\|^q \tag{1}$$

where C_j is defined as the center of the jth cluster and is the center nearest to the data object d_i, while the variable n denotes the number of elements in the data set, and q is an integer that defines the nature of the distance function ($q = 2$ for Euclidean distance) as discussed in Ahmad and Dey (2007).

The partitional clustering algorithm starts with an initial dataset partition and iteratively assigns data points or patterns to clusters to reduce the square error. A set of K seed points well separated from each other can be chosen randomly from the pattern matrix to be the initial partition. Nagpal (2013) noted that good seed points could be determined if the selected initial points are from existing data objects and sufficiently distanced from one another. As the number of clusters increases, the square error tends to reduce, and this minimization can only be achieved for a fixed number of clusters.

In some algorithms, the square error criterion function in partition clustering makes the generated K numbered clusters as compact and separated as possible. It is less computationally demanding than other criterion functions (Jain and Dubes, 1988). Because square-error-based algorithms can converge to local minimal, different initial partitions can produce varying clusters as output, especially if the initial points

are not well separated (Nagpal, 2013). According to (Jain and Dubes, 1988), partitional techniques are frequently used in engineering applications where single partitions are most important and appropriate for efficient representation and compression of large databases. It has also been observed that partitional algorithms are preferred in pattern recognition due to the nature of the available data (Jain, 2010). The partitional clustering method is a local search technique (Khaled et al., 1995) and local convergence. Therefore the optimal global solution cannot be guaranteed (Sanse and Sharma, 2015a).

Partitioned-based clustering is an NP-hard optimization problem with the standard approach of finding an approximate solution (Harshada et al., 2015). Jain et al. (1999) stated that "the combinatorial search of the set of possible labelings for an optimum value of a criterion function is computationally prohibitive". As a result, the typical partitional clustering algorithm runs several times with varying starting partitions so that the one that gives the best clustering output of all the runs is chosen as the optimal solution Jain et al. (1999). One major disadvantage of Partitional Clustering algorithms is the need for predefined user values for parameter k, which is usually non-deterministic (Suganya et al., 2018; Jain et al., 1999). This arbitrary choice of clusters centroid leads to wrong clustering output (Oyelade et al., 2016). Clustering Algorithms based on the partitional methods usually generate clusters of approximately similar sizes because data objects are permanently assigned to the nearest centroid, which invariably results in incorrect cut borders between clusters (Harshada et al., 2015). The partitional clustering method has been noted for having biasedness to spherically shaped clusters and that it is also unable to handle highly connected clusters and a high-dimensional dataset (Oyelade et al., 2016).

The partitional Clustering algorithm can be categorized based on the various techniques adopted in generating the clusters and the nature of the resultant clusters produced. These include Hard/Crisp Clustering, Fuzzy clustering, and Mixture Resolving Clustering.

4.2.1. Hard/Crisp CLustering

Each data object belongs to only one cluster in a hard or crisp clustering algorithm. The clustering methods under this category include Graph-theoretic clustering, Density-based clustering, Model-based clustering, Subspace clustering and Miscellaneous Clustering.

4.2.1.1 Graph-theoretic clustering A Graph structure is a data structure made up of nodes and edges connecting the nodes. A graph can model relationships between features of data objects and list important, relevant features during data analysis. In graph-theoretic clustering, clusters are represented in graphs (Saxena et al., 2017). The data objects are represented as nodes that are connected as edges. The edges reflect the proximities between pairs of data points (Xu and Wunsch, 2005). Nodes are divided into clusters so that the edge density across clusters is smaller compared to edge density within clusters (Saxena et al., 2017). Edges whose clustering length (weight) is substantially larger than the nearby edges' average is termed inconsistent. Nodes are grouped into clusters based on the graph topology so that the output clusters are characterized by high intra-connectivity/homogeneity and low inter-connectivity/homogeneity among the generated clusters. Representing clusters in graphs is convenient, but it is not robust in handling outliers.

Graph theory can be used to represent both hierarchical and non-hierarchical clusters. Graph method that directly deals with connectivity graphs can be used for linkage metrics-based hierarchical clustering when the connectivity $N \times N$ matrix is sparse (Jain and Dubes, 1988; Berkhin, 2001). This clustering method uses some topological properties in graphs to build clusters from a network of data objects. Finding the maximally connected subgraphs in a graph structure is the same as the problem of single linkage hierarchical clustering. In like manner, finding maximally complete subgraphs in a graph structure is equivalent to the total linkage hierarchical clustering (Jain and Dubes, 1988). The k-nearest-neighbor graph model was used to develop

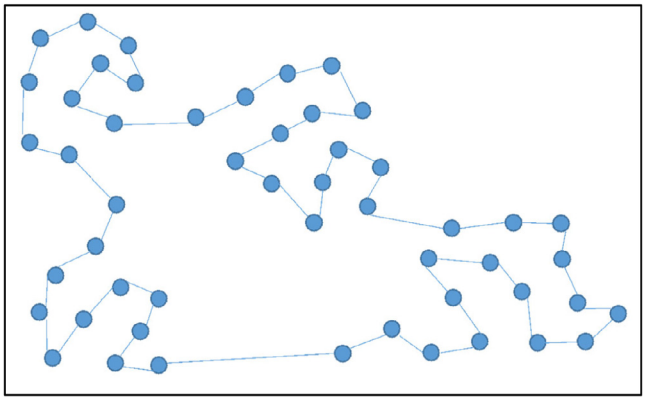


Fig. 4. Initial dataset represented as an undirected graph.

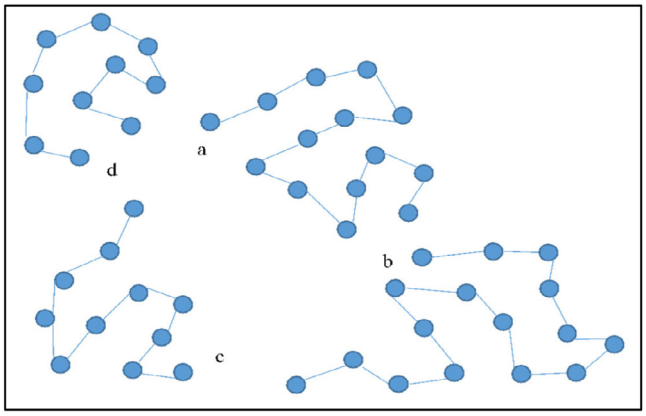


Fig. 5. Resulting clustered subgraphs with cuttings at points a, b, c, d.

Chameleon (an agglomerative hierarchical clustering algorithm) (Xu and Wunsch, 2005). Using the k-nearest-neighbor graph approach, Chameleon constructs a sparse graph with each data object representing a vertex of the graph with an edge existing between pairs of vertices. The weight of each edge indicates the similarities (distance) between the corresponding vertices. The k-nearest neighbor graph is partitioned into several relatively small sub-clusters using a graph partitioning algorithm in such a way as to minimize the weight of the edges to be cut. The clustering eliminates edges whose vertices are not within the k closest points concerning each other and uses an agglomerative hierarchical clustering algorithm to merge similar sub-clusters.

Another graph representation of hierarchical clustering is the Delaunay triangulation graph (DTG) that uses a hypergraph where more than two vertices are connected to an edge creating the hypergraph structure (Cherng and Lo, 2001). Zahn's clustering algorithm (Zahn, 1971) is an example of graph theory for non-hierarchical clustering. Uneven edges in minimum spanning trees are detected and discarded in the bid of connecting components as clusters (Jain and Dubes, 1988). However, there is a need for the cluster shape's pre-knowledge to select the proper heuristic to identify irregular edges. Cluster Identification via Connectivity Kernels (CLICK) is an example of a graph theory-based clustering algorithm. In CLICK, the minimum weight division is performed on the graph to generate clusters (Dongkuan, 2015; Sharan and Shamir, 2000). Specifying suitable parameters and criterion properties lead to some practical difficulties to be addressed (Jain and Dubes, 1988). According to Jain and Dubes (1988), "no theory exists for choosing among the various properties of graphs to select the best clustering method for a particular application". Figs. 4 and 5 show examples of graph-theoretic clustering approaches.

4.2.1.2 Subspace clustering Subspace clustering is an extension of the traditional clustering algorithm with the primary aim of finding clusters in different subspaces in which a dataset exists. It is often better to use the subspaces in which a dataset exists for its description instead of describing a large dimensional dataset as a whole (Parsons et al., 2004). This way, the subspace clustering technique helps discover hidden knowledge in such a sizeable dimensional dataset. Clusters existing in multiple overlapping subspaces are easily identifiable using subspace clustering. In subspace clustering, redundant and irrelevant dimensions are removed using feature selection, leaving only the relevant dimension that the clustering algorithm uses to find the clusters in the dataset.

The subspace clustering algorithm is categorized into two subsections using the clustering algorithm search strategies: the top-down and bottom-up approaches. The bottom-up subspace method uses an APRORI style approach to leverage density's downward closure property to reduce search space. The density's downward closure property's idea is those dense units exist in k-dimension, then dense units in (k-1) dimension projections. Based on this, the bottom-up method creates a histogram for each dimension and selects dimensions whose density is above a given threshold. Examples of bottom-up subspace clustering include CLIQUE (Agrawal et al., 1998), ENCLUS (Cheng et al., 1999), MAFIA (Goil et al., 1999), CBF (Chang and Jin, 2002), CLTree (Cheng et al., 1999), DOC (Procopiuc et al., 2002).

In the top-down subspace clustering, an initial approximation of the clusters in the whole feature space with equally weighted dimensions is first found. In the next step, a weight is assigned for each size in each cluster using a sampling technique to improve the algorithm's performance. Clusters formed using this approach forms partitions of the given dataset, with each instance of the data object belonging to exactly one cluster. There is a need to specify the number of clusters and the subspace's size ahead of time, which is a bottleneck for the approach. Parameter tuning must be performed to achieve a meaningful result. Dealing with outliers in the dataset is another challenge in this approach. PROCLUS (Aggarwal et al., 1999), ORCLUS (Aggarwal et al., 2000), FINDIT (Woo and Lee, 2002), COSA Friedman and Meulman, δ-Clusters (Yang et al., 2002) are examples of clustering methods that use this subspace clustering approach.

4.2.1.3 Density-based clustering In density-based clustering, dense regions in pattern space separated by low pattern density regions are viewed as clusters in the pattern space. These regions with high density called modes are associated with a cluster center, while the objects in the sparse areas separating the clusters are considered as noise and outliers (Harshada et al., 2015). The data points are then added to clusters with the closest center. A histogram is constructed by dividing the pattern space into non-overlapping regions to identify the pattern space modes. The regions with high-frequency counts form the potential modes and the histogram structure's valleys as the boundaries between the clusters. The major concern in using a histogram to measure the density function is that the pattern space must be large enough to identify the sections (Jain and Dubes, 1988). Furthermore, clusters that are small in size are usually very noisy because they cannot be adequately defined. In contrast, enormous clusters cannot properly define the cluster properties because of the member patterns' varied properties. It is also difficult to locate the precise values for the peak and valleys in the histogram.

Several works proposing the general concept for mode identification has been reported in the literature (Jain and Dubes, 1988). This clustering method has been used extensively in engineering, mostly in remote sensing applications (Wharton, 1983). In some other cases, clusters are formed based on the data points density within a region. Data points are added to the cluster until the neighborhood's density is less than a given threshold. In this case, a cluster in the neighborhood of a given radius must contain a minimum number of objects concerning the specified threshold. Generation of the cluster this way enables the building of clusters with arbitrary shapes. Outliers or noisy data points are naturally eliminated. Examples include Density-Based Spatial Clustering of Applications with Noise (DBSCAN), ordering points to identify the clustering structure (OPTICS), DENsity-based CLUstEring (DENCLUE). The DBSCAN has a well-defined cluster model with fairly low complexity (Harshada et al., 2015). OPTICS solved the DBSCAN's problem of choosing an appropriate value for the range parameter producing a hierarchical output similar to linkage clustering (Harshada et al., 2015).

The use of the spatial index in finding data point's neighborhood has been reported as improving the complexity of the model from $O\left(n^2\right)$ to $O(nlogn)$ compare with other methods (Nagpal, 2013). The density-based clustering method is reported as resistant to outliers, insensitive to data object ordering, ability to form arbitrary shape clusters, and no need for pre-stating the number of clusters (Sanse and Sharma, 2015b). However, they are not ideal for large data sets due to dimensionality. The low-density areas as noise make the algorithms based on this clustering method unable to detect intrinsic cluster structure common in real-life data. There is also the problem of cluster border detection because there is the need to have data point's density drop to show the demarcation between clusters (Harshada et al., 2015). Fig. 6 presents the clustering pattern of the density-based clustering method.

4.2.1.4 Model-based clustering In model-based clustering, data are assumed to be generated by an underlying probability distribution or a model (Fraley and Raftery, 1998). Each component of the distribution represents a different cluster. The principle is to recover the model and use it to determine the data points that satisfy the generated model or the probability functions in building clusters of similar data points. Model-Based clustering seeks to optimize the fitness of the predefined model concerning the given data. Since clusters are generated using the given data point, the total number of clusters present can be automatically generated to identify outliers easily. In Model-based clustering, a mixture model is used in representing data, and the components of the model correspond to the different clusters.

Fraley and Raftery (1998) reported two ways for formulating models for the composition of clusters: the classification likelihood approach and the mixture likelihood approach. Model parameters can be found using the Maximum Likelihood Estimation (MLE) criterion (Sanse and Sharma, 2015b) as well as the Bayesian information criterion (BIC) (Fraley and Raftery, 1998; Dasgupta and Raftery, 1998; Mukerjee et al., 1998). The BIC can also determine between two clusters the closest to which a data point will be assigned (Campbell et al., 1997). Data clustering uses two major approaches using this method: the statistical approach and the neural network approach (Sanse and Sharma, 2015b). Examples of the Model-Based Clustering method includes EM (Expectation-Maximization) (Fraley and Raftery, 1998; Dempster et al., 1977; Mclachlan and Krishnan, 1997), COBWEB, SOM. The Expectation-Maximization algorithm for maximum likelihood can determine the partition. A parametric mixture distribution for a random vector A such that A can be written as

$$f(\vartheta) = \sum_{b=1}^{B} \pi_b f_b\left(a|\theta_b\right) \tag{2}$$

where $\pi_b > 0$; such that $\sum_{b=1}^{B} \pi_b = 1$ are regarded as the mixing proportions. The $f_b\left(a|\theta_b\right)$ is the bth component density with the parameter vectors of the distribution represented as $\vartheta = \left(\pi, \theta_1, \ldots, \theta_b\right)$ with $(\pi = \pi_1, \ldots, \pi_b)$. The $f(\vartheta)$ is called the B-component finite mixture density. $f_1\left(\theta_1\right)$, $f_1\left(\theta_1\right), f_2\left(\theta_1\right), \ldots, f_b\left(\theta_1\right)$ represent the distribution components, that is, the clusters of the parametric mixture distribution. The distribution components are the same type for all the b. Shekar et al. (1987) proposed a knowledge-based clustering scheme by introducing the notion of conceptual cohesiveness as precise enough to be adopted for semantic grouping of related objects based on cohesion forest knowledge structure. The authors presented a set of axioms that should be satisfied to give meaning to the generated clusters.

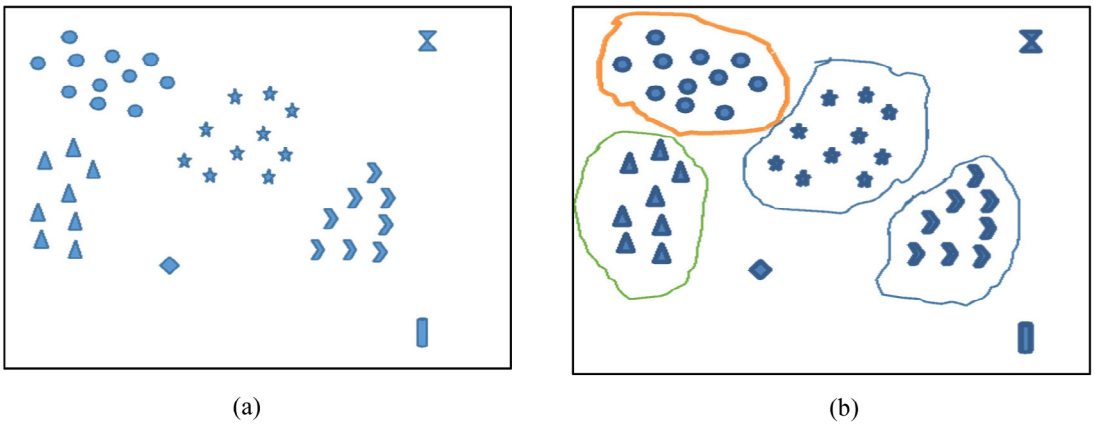


Fig. 6. (a) Initial dataset (b) Density-based clustering.

4.2.1.5 Search-based clustering Search-based clustering algorithms are nature-inspired metaheuristic approaches, termed automatic data clustering algorithms. They spontaneously determine the structure and number of clusters in a dataset without prior information on the dataset's attributes values (Aliniya and Mirroshandel, 2019). They emerge as a solution to the need to provide the traditional clustering algorithms, a priori information (Agarwal, 2011) on the number of clusters generated (Ezugwu, 2020a). The need to provide this vital information usually impose some additional computational burdens or requirements on the relevant traditional clustering algorithms (Ezugwu, 2020a). Determining the best estimate of cluster number is a fundamental problem in cluster analysis, and it is tagged 'automatic clustering problem' (José-García and Gómez-Flores, 2016). This problem becomes more pronounced for real-world data clustering analysis characterized by high density and dimensionality datasets. The lack of prior domain knowledge makes it difficult to choose appropriate cluster numbers, especially in datasets with many dimensionalities with widely varied cluster shape, size, density and sometimes overlapping. It is a profoundly difficult task to determine the optimal number of clusters for such data sets, so pre-identifying the number of clusters for a data clustering algorithm is not easy.

Automatic clustering techniques where such a requirement is not needed become a better option for real-world data sets with high density and dimensionality. Automatic clustering algorithms produce the same results as the traditional clustering technique without supplying any background information concerning the datasets (Aliniya and Mirroshandel, 2019; Jain and Dubes, 1988; Agrawal et al., 2005; Ezugwu, 2020a). It has also been found appropriate for handling automatic identification and classification of unlabeled data points in real-world datasets, which is evidently difficult and almost impossible manually. Automatic clustering algorithms have a higher possibility of obtaining optimal global solutions, unlike the traditional clustering algorithms that are mostly local search algorithms whose solutions are influenced by the initial starting points. They cannot guarantee global optimality except linear and convex optimization (Ezugwu, 2020a). Aside these, the nature-inspired clustering algorithms have demonstrated more flexibility in handling clustering problems in various fields than the traditional clustering algorithms that are mostly problem-specific and lack continuity (Agarwal et al., 2011). With the main aim of clustering algorithm as having the ability to generate clusters that exhibit the characteristics of reduced intracluster distance and increased inter-cluster distance, the automatic clustering algorithms treat clustering problems as optimization problems with a focus on the minimization of the dissimilarity within a cluster and maximization of dissimilarity between clusters (Ezugwu, 2020a; Kuo et al., 2014).

As an optimization problem, finding an optimal solution for a clustering problem is classified as NP-hard when the number of clusters is more than three; that is $k > 3$ (Falkenauer, 1998). Thus, clustering tasks for moderately sized problems could be computationally prohibitive. This makes most metaheuristic approaches suitable for finding solutions to data clustering problems (Kuo et al., 2014), and the metaheuristic search algorithms became the most applied techniques for implementing automatic clustering algorithms (José-García and Gómez-Flores, 2016). Nature-Inspired metaheuristic algorithms are practically designed to handle high-dimensional and complex real-world problems (Ezugwu et al., 2020b). Moreover, their higher heuristic search capability makes them look for the most promising (optimal) solution while balancing intensification and diversification in the search. Also, while searching for optimal solutions, they ensure that the generated solution gets into unpromising regions within the search space are avoided. These natured inspired metaheuristics algorithms have solved a wide range of continuous and discrete combinatorial optimization problems, particularly the GA, DE, PSO, FA and IWO (Ezugwu, 2020b). The automatic clustering algorithms are superior in performance compared with the traditional clustering algorithms in terms of convergence speed and their ability to produce good quality solutions

Some of the nature-inspired algorithms that have been deployed as search based clustering algorithms include Genetic Algorithm (GA) (Jain and Dubes, 1988; He and Tan, 2012; Doval et al., 1999), Differential Evolution (DE) (Paterlini and Krink, 2006; Suresh et al., 2009), Artificial Bee Colony Optimization Algorithm (ABC) (Kuo et al., 2014; Su et al., 2012), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) (Izakian et al., 2016; Das and Roy, 2008), Invasive Weed Optimization (IWO) (Chowdhury et al., 2011), Symbiotic Organisms Search (SOS), Bacterial Evolutionary Algorithm (BEA) (Das et al., 2009), Variable Neighborhood Search (VNS), Firefly Algorithm (FA) (Senthilnath et al., 2011) and Tabu Search (TS) Algorithm.

The metaheuristics-based clustering algorithms can be classified into the Evolutionary and the Swarm intelligence metaheuristics algorithms. The GA and DE come under the Evolutionary group, while the rest fall under the Swarm intelligence group. These two broad classes of algorithms have common design steps: the starting point of random initializing population size, then identifying suitable candidate individuals representing the choice solution (Ezugwu et al., 2020a). This is achieved by evaluating the candidate's members of the initial generation. The choice solutions are then used to generate a new population by modifying the individual-specific variation operators. The second and third steps are repeated iteratively, and an update is made concerning which candidate individual is best fitted in terms of the defined objective function of the problem. The best candidate's choice is achieved by comparing the current generation solution with the previous generation solution, and precedence is given to the current best solution. The subsequent sections will report reviews of different research involving the application of various nature-inspired algorithms to clustering problems.

(a.) Evolutionary algorithm

i. Genetic Algorithm-Based Clustering Techniques

Genetic algorithm is a single objective evolutionary computation algorithm that has been used for automatic clustering. Holland developed the algorithm in the early 1970s (Holland, 1975). Its idea stemmed from Charles Darwin's principle of evolution by natural selection. In genetic algorithms, some fundamental genetic ideas are borrowed and artificially used in constructing robust search algorithms with minimal problem information (Sheikh et al., 2008). The search is performed in large, complex multimodal landscapes providing a near-optimal solution for the search problem's stated objectives or fitness function. In GA-based clustering techniques, the capability of GA is applied to evolve the proper number of clusters and provide appropriate clustering (Sheikh et al., 2008).

The search space parameters are represented as strings called chromosomes. A combination of cluster centroids encodes them, and a collection of them forms the algorithm's population. At the first generation, a random population representing different search space solutions is created at the initial stage. Each chromosome has an objective and fitness function associated with it which measures the degree of goodness of the chromosome. Using the principle of survival of the fittest, the best fit among existing chromosomes is selected to 'birth' the next generation of chromosomes through the biologically inspired operators: the crossover and mutation operators. The selection, crossover and mutation operation are iteratively repeated for a given number of generations or until a stopping criterion is met (Goldberg, 1989). In GA-based clustering techniques, the selection operators control the search direction, while the crossover and mutation operators generate new regions for search.

Several research efforts in developing GA-based clustering algorithms have been reported in the literature. Krovi (1992) investigated the potential feasibility of the use of GA for the purpose of clustering. Krishna and Murty (1999) proposed Genetic K-Means Algorithm (GKA), a novel hybrid GA to find a globally optimized partition of a given dataset into a specified number of clusters using the K-Means algorithm to solve the problem of expensive crossover operations. The K-Means operators were used as search operators in place of the crossover. GKA searches faster and converges to the global optimum. It minimized total within-cluster variation (TWCV) (Krishna and Murty, 1999). The Fast Genetic K-Means Algorithm (FGKA) by Lu et al. (2004a) is another GA-based clustering algorithm inspired by GKA featuring several improvements over it. Other GA-based clustering algorithms include Incremental Genetic K-Means Algorithm (IGKA) (Lu et al., 2004b), GA-clustering (Maulik and Bandyopadhyay, 2000), Genetically Guided Algorithm GGA (Hall et al., 1999). The study conducted by the following authors: Sheikh et al. (2008), José-García and Gómez-Flores (2016); Ezugwu 2020a provides other references on further works on GA-based clustering. Based on a different encoding scheme for GA-based clustering, José-García and Gómez-Flores (2016) discussed four different categories of automatic clustering based on GA: the Centroid-based encoding of variable length, the Centroid-based encoding of fixed length, the Label-based encoding, and the Binary-based encoding.

(b.) Swarm Intelligence Algorithm

i. Ant Colony Optimization Clustering Algorithm

The ACO algorithm is a stochastic metaheuristic for combinatorial optimization (Dorigo and Stützle, 2004) classified under Swarm Intelligence (SI). SI is a category of artificial intelligence paradigm inspired by the study of emergent behavior in decentralized, self-organized systems. SI methods aim to imitate the behavior and apply it in finding solutions to hard computational problems. They are credited with having simple design capability, scalability and robustness. There are other algorithms based on the natural swarm behavior, such as bee colony, flocks of birds, and schools of fish. However, the ACO is categorized under ant-based techniques under SI. The Ant-based clustering methods are directly modeled on ant's social behavior (Dorigo and Stützle, 2004). It is the most widely used group of swarm-based clustering algorithms. There are two major approaches to ant-based clustering: those that directly mimic ants' behavioral nature and those that are less directly inspired by nature. The first groups considered gathering items and occasional sorting activities observed in the nest and brood care of ants (Deneubourg et al., 1991). The behavior is directly imitated in the clustering of abstract data where the clustering objective is implicitly defined. The second group, which is less directly inspired by nature, handles clustering tasks as an optimization task using the ant-based optimization method to generate good or near-optimal clusters. The second group has the advantage of explicit specification of the objective function, offering a better understanding and prediction of the clustering performance (Handl and Meyer, 2007). ACO clustering algorithm falls under this second group. It is inspired by the foraging behavior of mass recruiting ants. The ants use pheromones to mark areas of promising forage and potential food sources (Handl and Meyer, 2007; Dorigo and Stützle, 2004; Dorigo et al., 1996). Runkler (2005) and Saatchi and Hung (2005) carried out some research on the ACO-based clustering algorithms. An ant-based clustering algorithm was also presented by Kanade and Hall (2004), which finds the adequate number of clusters and initializes the fuzzy c means algorithm. Handl et al. (2006) worked on adaptive time-dependent transporter ant for clustering.

ii. Particle Swarm Optimization

The PSO is another general-purpose optimization metaheuristic algorithm inspired by the collective behaviors of unsophisticated agents that interact locally among the neighboring individuals and their environment to cause more complex valuable behavior for solving optimization problems (José-García and Gómez-Flores, 2016; Engelbrecht, 2005). Kennedy and Eberhart introduced the PSO as a population-based search algorithm where individual population members are grouped into a swarm. During optimization, the swarm of particles moves cooperatively in the defined region representing the objective function, with each particle representing a complete solution. Each particle moves in response to the influence of forces that attract it to a good position in the search space which previously has been explored by other swarm members or itself. According to José-García and Gómez-Flores (2016), "the particles explore the search space by adjusting their trajectories iteratively according to self-experience and neighboring particles". Several particles with randomly assigned velocities are initially placed at random positions in the search space at the search's commencement. Each particle will evaluate the objective function at its position at every iteration, updating its position, velocity, and memory for its individual best position (Handl and Knowles, 2007).

Omran first introduced the application of PSO for solving clustering problems in 2002. The algorithm used a fixed number of clusters, and then PSO was used in searching for the optimal centroids of the clusters and each data point assigned to the closest centroid. The work was further extended in their presentation of the DCPSO — dynamic clustering approach based on PSO (Omran et al., 2002; José-García and Gómez-Flores, 2016). Another PSO-based segmentation algorithm for automatically grouping image pixels into different regions was proposed by Das et al. (2006). Other researches on PSO based clustering algorithm can be found in (Qu et al., 2010; Ouadfel et al., 2010; Cura, 2012; Kuo et al., 2012, 2014; Van der Merwe and Engelbrecht, 2003; Cui et al., 2005; Cui and Potok, 2005; Kanungo et al., 2000). It has been observed that PSO-based clustering algorithms give excellent results in quality clustering to find the correct cluster number. The PSO basic algorithmic form is characterized by extreme simplicity. It is mostly used to optimize the functions of continuous variables.

iii. Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) is a swarm based metaheuristic optimization algorithm proposed by Mirjalili and Lewis in 2016 (Mirjalili and Lewis, 2016). The WOA was inspired by the social behavior of humpback whales with reference to their foraging behavior termed bubble-net hunting strategy. In hunting school of krill or small fishes close to the ocean surface, the humpback whales use two maneuvers: the 'upward-spirals' and the 'double-loops'. In the first maneuver,

a '9' shaped path is created around the prey from a depth of 12 m by the humpback whale and start to swim up towards the surface. The prey is captured in the second maneuver in three behavioral patterns called: the coral loop, lobtail and capture loop (Goldbogen et al., 2013).

Nasiri et al. (2018) proposed the Whale Clustering Optimization Algorithm based on the humpback whales' foraging behavior for data clustering. The advantages of WOA, among which include, low number of parameters and lack of local optimal entrapment, were harnessed in the proposed clustering algorithm. The main goal was to use the WOA for a complete search to cluster unlabeled data for better clustering results using a simple solution. Soppari and Chandra (2020) used an optimized clustering approach to develop an effective framework for digital watermarking. They combined Least Favorable-based Whale Optimization Algorithm (LF-WOA) with optimized Fuzzy-Cmeans (FCM) for selecting the initial centroid to identify regions for watermarks insertion in digital watermarking. More recent literature regarding the WOA-based clustering algorithm includes (Reddy and Babu, 2019; Rahnema and Gharehchopogh, 2020; Jadhav and Gomathi, 2018).

iv. Crow search algorithm

The Crow Search Algorithm (CSA) is a population-based metaheuristic optimizer inspired by the intelligent behavior of crows developed by Askarzadeh (2016). The crows are distributed widely and are considered among the world's most intelligent birds. Crows stores excess food in certain places, which are later retrieved when needed. They are known to be greedy, following each other to obtain better food sources. A crow watches and observes the location where other birds hide their food to steal it. It also takes precautionary steps of moving its own hiding place to prevent others from stealing its food. If a crow detects that another one is following it, the crow goes to another position of the environment to fool the one following it.

Lakshmi et al. (2018) combined CSA with K-means algorithm to improve the performance of K-means algorithm to achieve global optimum clustering solution. The CSAK algorithm was used to find the optimum solution for the initial centroids for the K-means algorithm. Balavand et al. (2018) also combined K-means with CSA for automatic clustering based on data envelopment analysis which measures the efficiency of the decision making units of the algorithm. In this case, the CSA carry out the clustering processing using the initial cluster centers generated by the K-means algorithm. Wu et al. (2015) proposed a hybrid clustering algorithm based on WOA and CSA, harnessing the advantages of the two algorithms with respect to their search strategy. In Anter et al. (2019), the CSA improved the Fast Fuzzy C-means algorithm for data clustering. The CSA generates the initial cluster centers for the fuzzy C-means algorithm for a more accurate cluster result. In the proposed algorithm, generating an optimal cluster center ensures the FFCM avoids getting stuck in the local minimal and improves computational performance.

v. Emperor Penguin Optimizer

The Emperor Penguin Optimizer is a bio-inspired metaheuristic algorithm introduced by Dhiman and Kumar (2018). It mimics the emperor penguin's huddling behavior for successful survival in the depth of the Antarctic winter. The huddling behavior is made up of four major steps: the huddle boundary generation, computation of the temperature around the huddle, calculating the distance and finding the effective mover. Ragaventhiran and Kavithadevi (2020)) adopted CSA as an optimizer in their proposed Frequent Pattern Mining in which Affinity propagation-based clustering is implemented in one of the five processes with the main objective of performing preprocessing to remove data redundancy. Furthermore, the authors validate the performance of their implemented method with previous approaches for succeeding metrics that are execution time, response time, load balancing rate, and scalability. The results obtained in the work revealed that the authors' proposed map-optimize-reduce mining technique on Hadoop achieved excellent results compared to existing literature results.

c. Other Algorithms

i. Variable Neighborhood Search

The Variable Neighborhood Search (VNS) is a metaheuristic algorithm that represents a flexible framework for a heuristic building to find approximate solutions to combinatorial and non-linear continuous optimization problems. It was proposed by Mladenovic and Hansen in 1997 (Mladenovic and Hansen, 1997). It is characterized by systematic exploitation of changes of neighborhood, finding the local minimum in the descent phase while escaping from the corresponding valley. The neighborhood structures are systematically changed while searching for an optimal (or near-optimal) solution. According to Brimberg et al. (2017), it is a proven heuristic framework for finding good solutions to combinatorial and global optimization problems. Compared with other metaheuristics, the VNS basic scheme and its variants are simple, requiring non or few parameters. The reasons for these characteristic behaviors of VNS are stated by Alguwaizani et al. (2011) as based on the following properties:

- Relativity of a local optimum to the corresponding neighborhood structure, that is, a local optimum relative to one neighborhood structure, is not necessarily a local optimum for another neighborhood structure.
- Consideration of global optimality with reference to local optimal in terms of all neighborhood structure. 'A global optimum is a local optimum with respect to all neighborhood structures' (Hansen and Mladenovic, 2018).
- The relative closeness of all or majority of local optimal to one another. In the words of Alguwaizani et al. (2011) 'Empirical evidence shows that all or a large majority of the local optima are relatively close to each other for many problems.

The VNS increasingly uses complex moves to find the local optimal in all the neighborhood structures. If the local optima found is poor, several neighborhoods are used. An increase in exploitation of the vicinity of the incumbent solution is also suggested (Hansen and Mladenovic, 2018).

The basic VNS algorithm, as with other metaheuristic algorithms, starts with a set of randomly generated initial solutions. This is followed by the random generation of a neighbor of the incumbent solution. This phase is called the shaking step. After this phase, the moving step is then performed where the local optimal and the incumbent solution are compared, and the incumbent solution is updated if the local optimal solution is better. These two phases are repeated until the maximum neighborhood number's termination condition is met.

The VNS and its variants have been used to solve many clustering problems. Alguwaizani et al. (2011) used VNS to solve the harmonic means clustering problem and reported that VNS compared favorably with solutions obtained from Tabu Search TS and Simulated Annealing SA Heuristics. The capacitated clustering problem was solved using two VNS based heuristics by Brimberg et al. (2019) and the performance accessed on benchmark instances from the literature. According to their report, all VNS procedures outperform the state-of-the-art in the stated problem. Consoli et al. (2019), Orlov et al. (2018); Mladenovic and Hansen (1997), Hansen (2005), Hansen and Mladenovic (2018), Hansen et al. (2009), Rozhnov et al. (2019), Hansen and Mladenovic (2001), Martins (2020), Carrizosa et al. (2013) are other research reports on using VNS and its variants for solving clustering problems.

Ros and Guillaume (2019), in related work on neighborhood search, proposed a clustering method referred to as Munec to address the challenge of new clustering algorithms in finding an appropriate number of clusters in complex datasets and demonstrate self-tuning capability and adaptiveness for input parameters in identifying acceptable solutions. Their algorithm adopted the nearest neighbor technique in group related data objects. Nearest neighbor data objects are merged without constraints at the beginning until the number of groups attains the maximum (at least two groups). Subsequent merging is then based on mutual neighbor groups with a similar distance between neighbors.

The experimental tests carried out using two-dimensional datasets revealed that Munec proved to match a ground truth target highly

effectively. More so, under the same input configuration, Munec can identify clusters of various densities, arbitrary shapes, and a large amount of noise (Ros and Guillaume, 2019).

ii. Tabu Search Algorithm

The Tabu Search (TS) Algorithm is an optimization problem-solving higher-level heuristic procedure designed to guide other methods in escaping the possibility of getting trapped in local optimal (Glover, 1990). It uses a local search procedure guided to avoid local optimal, rejecting already visited points (kept on the tabu list) in search space (Batres, 2012). A thorough information search is enabled in tabu search using a flexible memory structure. This aids the algorithm in strategically constraining and freeing the search process with varying period memory functions for intensification and diversification. According to Glover (1990), "the form of guidance provided by tabu search is highly flexible and often motivates the creation of new types of moves and evaluation criteria to take advantage of its adaptability to different problem structures and strategic goals". TS has been credited with finding solutions superior to the best solutions of other alternative methods for various problem settings. Implementing the TS algorithm is easy, and it can handle additional considerations. For instance, constraints that are not included in the original formulation of the problem are given consideration. TS algorithm was applied to subset clustering problems by Glover et al. in 1985 with a solution obtained in less than one minute on a V77 minicomputer (Glover, 1990). As mentioned earlier, three basic themes can be identified in the development of the TS algorithm with its core embedded in the short-term memory process:

- Usage of flexible attribute-based memory structure
- An associated control mechanism for employing the memory structure and
- Incorporation of memory functions of different time spans from short to long term.

In recent times, there have been several reports on using TS algorithms to solve various clustering problems. Cao et al. (2015) presented a TS algorithm for solving cohesive clustering problems in various business applications. They introduced an objective function for generating as pure as possible clusters, maximizing the intracluster similarity as much as possible. They employed the intensification and diversification strategies of tabu search to enhance the clustering outcome. Sung and Jin (2000) combined the tabu search algorithm with complimentary packing and releasing procedures for solving the clustering problem. Lu et al. (2018) proposed a tabu search-based clustering algorithm and its parallel implementation on Spark. Their design was adapted to alleviate the challenges associated with big data applications taking advantage of the parallel processing based on the Spark framework. They found the system superior to other similar systems in terms of scalability, accuracy, and effectiveness. Other TS algorithm applications for solving clustering problems can be found in Kharrousheh et al. (2011), Xia et al. (2018), Yaghini and Ghazanfari (2020) and Ibrahim et al. (1994).

4.2.1.6 Square error clustering The square error clustering method is a partitioning clustering method that assigns data points into a specified number of clusters based on the sum of square error criterion functions. The squared differences between each data point and the estimated center value for each stated group have been divided into the data point. In cases where the sum of squared error for a group of data objects is equal to zero, the cluster's data points are identical (very close).

The formula for Sum of Square Error is:

$$Sum\ of\ square\ error = \sum_{i=1}^{n} \left(x_i - \underline{x}\right)^2 \tag{3}$$

where n represents the number of data points and x_i represents the ith data point in the group and $\underline{x}$ is the center object relative to the group. The k-means clustering algorithm is the best know squared error-based clustering algorithm (Xu and Wunsch, 2005).

i. K-Means Clustering

The K-Means clustering algorithm is a centroid-based partitioning technique in which data objects are distributed into a specified number of k clusters. The distribution is done through the use of an objective function which accesses the quality of the partition, ensuring that the similarities of objects within a cluster (intra-cluster similarity) is higher compared with objects in another cluster (inter-cluster similarity). K-Means clustering is a centroid-based technique, and it uses the mean to represent the centroid of a cluster. The centroid of a cluster is a measure of the center point of the cluster. Specified k numbers of data points/objects are randomly selected from a set of the existing data points as the representative center for k clusters. The Euclidean distance between the remaining data points and each assumed center point is then iteratively measured. The obtained value assigns the data point to the cluster with the smallest distance. The intracluster similarity is improved each time a new data point is given to the cluster by computing a new mean using the objects previously assigned to the clusters. The new mean is then used to reassign the data objects. This procedure is repeated severally until stability is achieved.

The sum of square function for the Euclidean distances produces compact and well-separated clusters. The K-Means algorithm tries to minimize the sum of the squared error criterion (Ezugwu, 2020a; Hartigan and Wong, 1979; MacQueen, 1967). The major problems identified with K-Means clustering algorithms include the problem of the initial definition of the number of clusters at the algorithm's onset. An efficient and universal method for determining the initial number of clusters and partition is not found. K-Means algorithm is reported to be very sensitive to initial centroid selection such that suboptimal solution may be produced when wrongfully chosen (Punit, 2018). Also, convergence to global optimum cannot be guaranteed. Using means as a centroid limits the K-Means algorithm's application to data objects with numerical variables (Xu and Wunsch, 2005). Not only these, but the K-Means algorithm is also sensitive to outliers (objects that are quite far from the cluster centroid are forced into the cluster, distorting the cluster's shape (Saxena et al., 2017). It works on the assumption that the variance of the distribution of each attribute is spherical and thus produces a roughly equal number of observations. Moreover, the memory space requirement is high and the number of iterations to obtain a stable distribution is unknown. Due to the simplicity of implementation and low computation complexity (Jain, 2010), the K-Means algorithm is still popular and widely used today (Ezugwu, 2020a).

Some research work extending K-Means has been reported. For example, the G-means (Hamerly and Elkan, 2004) and the X-means algorithms (Pelleg, 2000). The sum of square function for the Euclidean distances for the K-Means algorithm is given as:

$$d_{ik} = \sum_{j=1}^{m} \left(x_{ij} - c_{kj}\right)^2 \tag{4}$$

where d_{ik} is the Euclidean distance, x_{ij} is the jth data point for ith cluster and c_{kj} is the centroid for the jth cluster.

The K-Means is arguably the most popular clustering method but is plagued with drawbacks such as poor scalability, sensitivity to initialization and outliers, assumed knowledge of cluster count, and local production rather than the global optimum. It is noteworthy to mention that the most recent extensions and improvements on the K-Means seek to advance the state-of-the-art in addressing these issues.

ii. K-MCI (K-Means modified cohort intelligence) Clustering algorithm

The K-MCI (K-Means modified cohort intelligence) is an efficient hybrid evolutionary data clustering algorithm that combines the K-Means algorithm with modified cohort intelligence (Krishnasamy et al., 2014). Cohort Intelligence(CI) is an optimization algorithm inspired by the natural and societal tendency of cohort candidates/individuals learning from one another. It was proposed by Kulkarni et al. (2013). In cohort intelligence, while observing every other candidate, each candidate tries to improve their behavior. The MCI is a modified cohort intelligence with improved accuracy and speed of convergence of the traditional CI. In K-MCI, the K-Means algorithm enhances the

candidate's behavior generated by MCI, annexing the advantages of the K-Means algorithm and that of the MCI. K-MCI converges more quickly with greater clustering accuracy without being trapped in the local optimum.

iii. ELM K-Means (Extreme Learning Machine K-Means)

In ELM K-Means, the extreme learning machine (ELM) method is incorporated into the K-Means clustering algorithm (Alshamiri et al., 2015). The ELM method functions in projecting the dataset into a high dimensional feature space, and the K-Means algorithm is used to cluster the dataset using the Euclidean distance in the feature space to measure the similarity between the objects. The ELM proposed by Huang et al. (2006) is a new learning algorithm that randomly generates hidden nodes for single hidden layer feedforward neural networks(SLFNs) and determines the output weight of the SLFNs analytically. ELM is credited with a meager computational cost for its operations and has been used in finding the solution to classification and regression problems.

iv. K-means based multiview clustering methods and K-means subspace clustering models

The generation of high dimensional data due to the social network's rapid development has been a significant challenge to the traditional K-means clustering generally tagged as the curse of dimensionality. Redundant features and noises in such data make efficient clustering of such data very difficult. The K-means based multiview clustering methods are developed to provide simple and efficient algorithms for accurately exploring shared information in multiview data. Zheng et al. (2018) proposed a robust discriminative multiview K-means clustering with feature selection and group sparsity learning. The proposed algorithm addressed the problem of extreme time consuming and sensitivity to outliers that is common with clustering of high-dimensional feature space. It efficiently handles the curse of dimensionality by using group sparsity constraints for selecting the most important views and the most relevant features.

In handling high dimensional data of a real-world application, using eigenvalue decomposition by existing K-means subspace clustering algorithm to find an approximate solution is less efficient. Moreover, their loss functions exhibit sensitivity to outliers or suffer small loss errors (Wang et al., 2019). A new adaptive Multiview subspace clustering method was recently proposed by Yan et al. (2020) for integrating heterogeneous data in low-dimensional feature space. Their work extended K-Means clustering with feature learning capability for handling high-dimensional data. Wang et al. (2019) developed a fast adaptive K-means (FAKM) type subspace clustering model embedded with a mechanism for flexible cluster indicator using an adaptive loss function.

According to Wang et al. the existing methods of combining subspace learning with K-means clustering still exhibit some limitations. These include: no thorough capturing of discriminative information in low-dimensional subspace, consideration of intrinsic geometric information is rare, and the vulnerability to noises of the optimizing procedure of a discrete cluster indicator. They proposed a robust dimension reduction for clustering with a local adaptive learning algorithm to address these limitations. The proposed algorithm adaptively explores the discriminative information by unifying K-means clustering with local adaptive subspace learning.

4.2.1.7 Miscellaneous clustering techniques **i. Time series Clustering**

A time series is a sequence of real numbers collected regularly in time, where each number represents a value. It is the simplest form of temporal data, which is naturally characterized as high dimensional and large data size (Chiş et al., 2009a,b; Antunes and Oliveira, 2001; Warrenliao, 2005; Rani and Sikka, 2012; Lin et al., 2004). Since time-series data feature changes at a function of time, they are classified as dynamic data. Each time series is made up of many data points, but at the same time, they can be seen as a single object. Time-series clustering is an aspect of temporal data mining research that provides useful information in various domains (Liao, 2005; Aghabozorgi et al., 2015; Wang et al., 2002; Das et al., 1998).

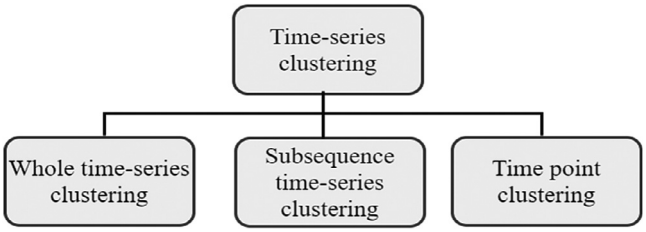


Fig. 7. Time-series clustering taxonomy (Zolhavarieh et al., 2014).

By clustering time-series data, diverse scientific areas have discovered patterns that data analysts have used to extract valuable information from complex and massive datasets (Aghabozorgi et al., 2015). According to Aghabozorgi et al. (2015), Chiş et al. (2009a,b), time series clustering as an exploratory data mining technique is the most used approach and has also been a subroutine in more complex data mining algorithms. The visual image representation of time series cluster structures helps users quickly understand the clusters, the anomalies, the structure of the data, and other datasets' regularities. Time series clusters have been used in finding answers to numerous real-world problems such as anomaly, novelty or discord detection (Keogh et al., 2002; Chan and Mahoney, 2005; Wei et al., 2005; Leng et al., 2009), recognition of dynamic changes in time series (He et al., 2011), prediction and recommendation (Sfetsos and Siriopoulos, 2004; Pavlidis et al., 2006; Ito et al., 2009; Graves and Pedrycz, 2010) and pattern discovery (Wang et al., 2002; Das et al., 1998). The enormous size of time series data requires being stored on disks during processing, resulting in an exponential decrease in the speed of the clustering process.

Time-series clustering can be classified into three (Aghabozorgi et al., 2015): the whole time series clustering, the subsequence clustering and the time point clustering. Performing clustering operations on many individual time series to similar group ones into a cluster is called the Whole-time series clustering. In this case, each time series is treated as an object, and the conventional clustering algorithm is employed. Subsequence clustering entails a clustering set of sliding window extractions of a single time series intending to find the similarity and differences among the extracted time windows. According to Chiş et al. (2009a,b), the subsequence clustering is a standard subroutine in rule discovery algorithm (Das et al., 1998; Fu et al., 2004; Halkidi and Vazirgiannis, 2001), indexing, classification algorithm, prediction algorithm (Ormerod and Mounfield, 2000; Popivanov and Miller, 2002) and anomaly detection (Steinback et al., 2002). The time point clustering (Aghabozorgi et al., 2015; Gionis and Mannila, 2003; Ultsch and Mörchen, 2005; Morchen et al., 2005) involves clustering of time point-based on the combination of the time point's temporal proximity and the similarity of the corresponding values. Fig. 7 presents a time-series clustering taxonomy.

There are different ways to cluster time series data recorded in the literature. These includes:

- Customize existing conventional clustering algorithms (Aghabozorgi et al., 2015; Warrenliao, 2005).
- Conversion of time series data into simple objects as input into conventional clustering algorithms (Aghabozorgi et al., 2015; Warrenliao, 2005).
- The use of multi resolutions of time series as input of a multi-step approach (Aghabozorgi et al., 2015)

Three different clustering time series data approaches were identified: feature-based, model-based, and shape-based approaches. The shape-based approach employs the conventional clustering methods to match two time-series shapes by a non-linear stretching and contracting of the time axes using a distance or similarity measure appropriate for the time series. The feature-based convert the raw time series into

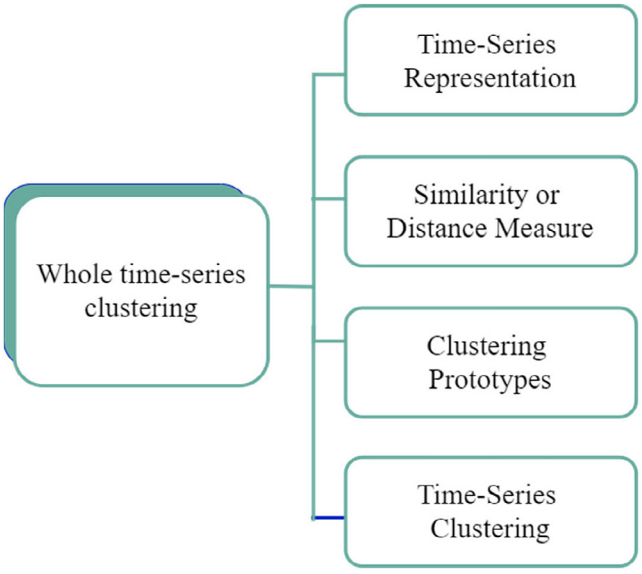


Fig. 8. Four Components of whole time series clustering.

feature vectors of lower dimension then apply a conventional clustering algorithm to extract the feature vectors. The model-based method involves transforming the raw time series into model parameters for each time series. Then, it applies a conventional clustering algorithm and a suitable model distance to extract model parameters. Model-based has scalability problems, and there is performance reduction when clusters are close to each other (Vlachos et al., 2004; Mitsa, 2009). The four components of whole time series clustering is depicted in Fig. 8.

Time Series clustering is made up of four major components (Aghabozorgi et al., 2015):

i. Dimensionality Reduction or Representation Method
ii. Distance Measurement
iii. Clustering Algorithm
iv. Prototype Definition

ii. Streaming Clustering

A data stream is an infinite and fast-changing massive sequence of multidimensional objects that continuously arrives rapidly over time (Mansalis et al., 2018). A data stream cannot be stored in memory or disk due to its infinite nature, so it is besieged with the constraint of a single pass over the data. Also, there is no control over the order of data arrival, and random access of objects is not possible. As a result of these constraints, the conventional clustering algorithms cannot sufficiently handle data streams' clustering problems (Gama and Gaber, 2007), hence the need for streaming clustering algorithms. Mansalis et al. (2018) identify three major challenges peculiar to streaming clustering algorithms: the evolving nature of data streams and the number of clusters and outliers handling. As a result of the evolving nature of data streams, Streaming clustering algorithms must continuously update extracted clusters to capture changes in underlying data. It will be highly restrictive if a fixed number of clusters is assumed for a data stream since it is infinite. There is a need to configure the number of clusters over time. Identifying outliers upon arrival of an object in a data stream is difficult because both the clusters and outliers are developing over time. Other three problems of data streams that are general to data processing activities which are still relevant to streaming clustering include the single pass constraints mentioned earlier, limited processing time, which require the fast response of processing algorithm, and the limited memory, which makes working with summary data essential when data stream processing is concerned.

In data streams, window models control the part contributing to the data mining patterns. Landmark Window model, sliding window model, Damped window model, and Tilted window model are the literature's various window models. Mansalis et al. (2018) provides more information on the various window models. St streaming clustering algorithms can follow two major directions during the clustering process: Online Clustering and Online–Offline clustering. In Online clustering, the stream clustering problem is viewed as a single pass clustering challenge.

Hence to maintain the clusters, a general adaptive strategy is adopted. The single clustering model adopted is maintained and updated as new data objects arrive from the stream. This way, cluster structure cannot be investigated at different time intervals. Aggarwal et al. (2003) introduced Online–offline clustering to overcome the problem in online clustering. The data statistics are maintained online during the online phase, while the actual clustering is performed on the statistics based on the user-defined temporal predicates during the offline phase. Different time windows can be selected for possible multiple cluster extraction in this method, thus providing flexibility in data stream exploration. As earlier stated, the infinite nature of the data stream makes storing the data impossible; hence summarization becomes necessary. According to Mansalis et al. (2018), the most common user summaries for clustering include Cluster features (CF), Micro-Clusters, Core-micro-clusters, Temporal cluster feature, Prototype array Grids, Coreset Tree.

Streaming Clustering Algorithm can be categorized into four based on the existing taxonomy of conventional clustering algorithms. The partitioning algorithms category produces spherical clusters that do not handle outliers. Criterion Optimization like the Sum of Squared Errors is used in the partitioning. Example of such algorithms includes Clustream (Stream Clustering) (Aggarwal et al., 2003), Stream Framework Algorithm (Guha et al., 2000), StreamKM++ (K-Means clustering algorithm for data streams) (Ackermann et al., 2012), SWClustering (Zhou et al., 2008). The density-based category includes HDenStream (Density-Based Clustering Algorithm for Heterogeneous Data Stream) (Lin and Lin, 2009), DenStream (Density-Based Clustering over an Evolving Data Stream with Noise) (Cao et al., 2006), MuDi-Stream (Multi Density data Stream), rDenStream (Density-Based Clustering Algorithm with outlier retrospect) (Xiong et al., 2009), HDDStream (density-based projected clustering algorithm for high dimensional data streams) (Ntoutsi et al., 2012), SDStream (Density-Based Data Streams Clustering over Sliding Windows) (Ren and Ma, 2009) and C-DenStream (a density-based clustering algorithm for data streams that includes domain information in the form of constraints) Ruiz et al. (2009). As it is with conventional density-based clustering algorithms, they can discover clusters of arbitrary shapes and identify outliers. The third category is the Grid-based category which includes DDStream (Zhou et al., 2008), DENGRIS (Density-Grid based Clustering Algorithm) Amini et al. (2012), D-Stream (Stream Data Clustering Based on Grid Density and Attraction) (Chen and Tu, 2007), PKS-Stream (Data stream Clustering Algorithm based on grid density and index tree PKS tree) (Ren et al., 2011) and MR-Stream (Density-Based Clustering of Data Streams at Multiple Resolutions) (Wan et al., 2009). The Model-based Algorithms are the fourth category that tries to fit a model to the data. SWEM (An EM-Based Algorithm for Clustering Data Streams in Sliding Windows) (Dang et al., 2009) is an example.

The Stream framework under the partition category works by breaking the data stream into batches of fixed sizes and applying the K-median clustering algorithm. The problem of memory limitation and single-pass constraint was adequately dealt with. However, the problem of noise and data aging persists with the single model being maintained over the stream, which is inadequate in describing the stream's whole evolution. A detailed description of each category and method can be obtained from the stated references for each technique.

iii. Mode Seeking Clustering Algorithm

Mode is a measure of central tendency that returns the value that occurs most frequently in a data set. The mode can be determined for qualitative and quantitative attributes, and it is possible that a single data set can have more than one mode. In mode-seeking clustering algorithms, estimated density functions generate clusters (Fukunaga and Hostetler, 1975; Comaniciu and Meer, 2002). These modes are the local maxima of the probability density functions. Mode-seeking clustering assigns cluster labels by associating data samples with the nearest modes (Sasaki et al., 2018). The number of detected modes automatically forms the number of generated clusters in the mode-seeking clustering algorithm. According to Duin et al. (2012c), the mode-seeking clustering can be considered an agglomerative approach where a density function is estimated for the dataset (running a mean-shift iteration initialized at every data point) and have each mode defining one cluster. In the clustering phase, to decide which mode an object belongs to, the density gradient from that object is followed until a mode is found. Objects that end up in the same mode belong to the same cluster. This procedure allows the number of clusters to be identical to the number of modes (Duin et al., 2012c; Carreira-Perpiñán, 2015). Duin et al. (2012c) discussed two-mode seeking procedures relative to the non-parametric density estimates used: the mean shift procedure (that uses the Parzen Kernel for mode seeking (Fukunaga and Hostetler, 1975; Cheng, 2002)) and the kNN mode seeking procedure which uses the K-nearest Neighbor estimator (Koontz et al., 1976; Kittler, 1976; Shaffer et al., 1979). The two procedures have a width parameter that influences the number of modes in the density estimate, and the consideration is on the clustering.

The Mean shift clustering is a mode-seeking clustering algorithm that initially considers all objects of a data set as candidates for cluster centers which are then updated iteratively towards the nearest mode of the estimated density by gradient descent. In Myhre et al. (2018), Comaniciu and Meer (2002), mode seeking is a prominent density-based clustering method represented mainly by the Mean shift algorithm. To some extent, the mode-seeking clustering algorithms do allow capturing nonlinear clusters because the density can adapt locally to the data. As stated earlier, the density estimate determines the number of clusters. Apart from these, it is also robust to outliers because an outlier is represented by its cluster and, based on its density value, can easily be thresholded away (Myhre et al., 2018).

The k-NN mode seeking procedure defines a pointer for every object with the highest density in its neighborhood. The density of every object is proportional to the distance to its kth Neighbor. The pointers are then followed to the object that points to itself because it represents a mode in the density as the objects with the highest density in its neighborhood. The k-NN procedure has been reported to be significantly faster than the mean shift algorithm and can handle larger datasets (in terms of large numbers of an object and high dimensions) compared with the Mean shift algorithm, which can handle large datasets for only low dimensional spaces (Duin et al., 2012a,b). This is because the mean shift algorithm has problems tracking the density gradient in high dimensions. From the conclusion (Duin et al., 2012c), the mode seeking clustering algorithm is presented as the most natural procedure for cluster analysis, but there is a need for the dataset to be sufficiently large enough before a good density estimate can be obtained.

iv. Multiview Clustering

The big data paradigm introduced multi-view data in more recent times, which are data observed from different views or generated from different sources (Yang and Wang, 2018). Multiview data exhibits its specific heterogeneous property while holding potential connections with others. The specific property of a particular view may be associated with a particular task of knowledge discovery with other different views containing information that is complementary to it, which may be exploited. Multiview clustering involves using advanced techniques that exploit the complementary and consensus information that falls across the multiple views from where the data is drawn. It provides a means of discovering the hidden power of knowledge embedded in such type of data. The early work on multiview clustering includes reinforcement clustering for multi-type interrelated data (Wang et al., 2003), two view version of EM-based and agglomerative algorithm (Bickel and Scheffer, 2004) and multiview version of DBSCAN (Kailing et al., 2004).

The inherent problem being addressed by multiview clustering is maximizing the clustering quality within each view while maintaining clustering consistency across the different views. Other challenges of multiview clustering include the successful handling of incomplete multiview data. Incomplete multiview data have some data objects that have no observation in some of the views or have only part of their features registered on the view.

Five categories of multiview clustering algorithms were discussed from the survey work on multiview clustering carried out by Yang and Wang (2018). These include multi-kernel learning, co-training style algorithms, multi-view subspace clustering, multi-task multi-view clustering and multi-view graph clustering. In multi-kernel learning, predefined kernels that correspond to different views are used. The kernels are then combined in a linear or non-linear order to improve clustering performance. The co-training style algorithms use a co-training strategy to treat multiview data. It uses prior knowledge or learning knowledge obtained from other views to bootstrap each view's clustering. This process is done iteratively with each view's clustering results tending towards each other to produce the broadest consensus that cut across all the views.

The multi-view subspace clustering assumes that all views share a unified representation. Based on this assumption, this unified representation that serves as input into a model for clustering is learned from all the views' subspace features and used for clustering. The methods under this category include the subspace learning-based method and the non-negative matrix factorization-based method. The multi-task multi-view clustering saddles each view with a single or multiple related tasks. The inter-task knowledge is transferred among the various views to exploit the Multiview relationship and the multitasking capability to improve the performance of the clustering process. The multi-view graph clustering applies a graph clustering algorithm or related algorithm such as spectra clustering on a sought-out fusion graph that cuts across all the multiview data views. This category of multiview clustering is further sub-divided into three methods based on the clustering method applied. These are the graph-based method, network-based method and spectral-based method.

The success and effectiveness of multiview clustering are based on two related principles: the complementary principle and the consensus principles. These two principles reflect the underlying assumptions employed in the clustering process and how such algorithms are modeled and operated. The complementary principle reflects the necessity of employing multiview for a comprehensive and accurate description of data objects. In contrast, the consensus principle handles consistency maximization across multiple distinct views based on the generalization error analysis proposed by Dasgupta et al. (2002).

According to Yang and Wang (2018), each category of the multiview algorithm has its pros and cons. For instance, in the co-training style, the clusters of different views are enhanced interactively through the information exchange, but the approach becomes intractable when the

view size exceeds three. The kernel-based has the advantage of the kernel but has high computation complexity. The interpretability of the multiview subspace method is straightforward, but it suffers from dependence on initialization parameters. The multi-view graph method harnesses the spectral graph theory and its advantages. However, it relies on the constructed affinity. In the multi-task multiview method, the approach enjoys the advantages inherent in multi-view clustering and multi-task cluster properties. Nonetheless, the research work in this area is still relatively new.

v. Deep Learning Clustering

Deep learning clustering methods use deep neural networks to learn clustering representations (Min et al., 2018). The optimizing objective of the deep clustering usually refers to as the loss function, has two parts: the clustering loss L_c and the network loss L_n. The network loss L_n learns the feasible features and also avoids irrelevant solutions while the L_c fosters the formation of feature points groups or become discriminatory. The loss function is given as :

$$L = \lambda L_n + (1 - \lambda) L_c \tag{5}$$

where $\lambda \in [0, 1]$ as a hype-parameter balances L_n and L_c. The use of deep neural networks for data clustering makes learning no-linear mappings possible, transforming data into a more clustering-friendly representation, eliminating the need for manual feature extraction or selection.

The similarity methods used in convectional data clustering methods cause poor performance when clustering high-dimensional data (Min et al., 2018). Methods of feature transformation and dimensionality reduction have been applied for mapping raw data into new feature space for generated data that existing classifiers can easily separate. However, the high complexity of the latent structure of data still poses a challenge to the existing clustering methods effectiveness. Data can be transformed into a more clustering-friendly representation using deep learning algorithms such as deep neural networks (DNN) because of its inherent highly non-linear transformation characteristic. T deep neural networks-based clustering methods have proven promises for effective and efficient clustering of real-world data (Aljalbout et al., 2018).

There are existing novel deep learning-based clustering methods that combine deep neural networks with clustering methods. According to Li et al. (2018), the various approaches can be categorized into two: one, the unified approach that optimizes the clustering objectives and the deep representation learning, and two, the sequential methods that apply clustering on the learned DNN representation. Xie et al. (2016) proposed a deep embedding clustering (DEC) method, which uses deep neural networks to simultaneously learn feature representations and cluster assignments. Its operations involve mapping from the data to a lower-dimensional feature space while iteratively optimizing a clustering objective. Li et al. (2018) used fully convolutional auto-encoders for learning image features as a base for a unified clustering framework for joint image representation and cluster centers learning. In the same vein, Yang et al. (2017) proposed a joint K-means clustering approach and dimensionality reduction with the dimensionality reduction accomplished by learning a deep neural network. Other methods include DEPICT — deep embedded regularIzed clustering (Dizaji et al., 2017).), VaDE — variation deep embedding (Jiang et al., 2016), CCNN-CNN-based joint clustering (Hsu and Lin, 2018) and DTAGnet — deep learning task-specific and graph-regularized network (Wang et al., 2016).

Pitchai et al. (2021) proposed brain tumor segmentation using deep learning and fuzzy K-means for magnetic resonance images. Artificial neural networks and fuzzy K-means were combined to segment the tumor locale in their work. In Huang et al. (2021), the hidden representation associated with different implicit lower-level attributes is learned using their proposed robust deep K-means model. Systematic taxonomy of clustering methods that used deep neural networks was presented by Aljalbout et al. (2018). Their taxonomy presentation was based on a comprehensive review of recent work.

4.2.2 Mixture Resolving Algorithms

The Mixture Resolving Algorithm or mixture-based algorithm assumes that a set of observed objects emanates from a mixture of instances from multiple probabilistic clusters. Therefore, a probabilistic cluster is chosen according to the cluster's probabilities to generate each observed object. A sample is then chosen according to the probability density function of the chosen cluster. The data set is assumed to be a mixture of a given number of different cluster groups with varying proportions during clustering. The mixture likelihood-based approach to clustering is model-based because the specification of each component density of observation is required in advance. Aitkin and Rubin (1985) stated that a statistical model to be used must be stated or known ahead in the clustering of samples from a population. With this overlapping relationship between the model-based clustering and mixture-based algorithm, it is possible to conduct estimation analysis and hypothesis testing of clustering methods based on mixture models using standard statistical theory. Marriott (1974), in support of this, had stated that the mixture likelihood-based approach 'is about the only clustering technique that is entirely satisfactory from the mathematical point of view'. It assumes a well-defined mathematical model, investigates it by well-established statistical techniques and provides a test of significance for the results.'. The determination of the most suitable number of clusters can be easily achieved in a Mixture-based algorithm because it has a clear probabilistic foundation (Berkhin, 2012). Mclachlan and Basford (1988) stated that providing an effective clustering of various data sets under various experimental designs is one of the mixture model's usefulness. However, the assumptions made regarding the data distribution are rather strong, and the computation complexity is high. Moreover, each cluster is viewed as a single simple distribution, constraining the cluster's shape (Grira et al., 2005).

4.2.2.1 Expectation maximization The expectation–maximization algorithm is a framework that employs two major steps in approaching the maximum likelihood of estimates of parameters in a statistical model; the Expectation Steps and the Maximization Step. In the expectation step, objects are assigned to clusters based on the probabilistic clusters' parameters and in the Maximization step, the new clustering or parameter that maximizes the expected likelihood is found. Given initial random values for the probabilistic distribution parameters such as the mean and standard deviation, the E-step and the M-step are then iteratively conducted until the parameter converges or the change is sufficiently small. During clustering, each object's probability belongs to each distribution is calculated, and the probabilistic distribution parameters are adjusted to maximize each cluster object's expected likelihood in the M-step. Several computations are required for each iteration of the EM algorithm. The product of the number of data points and the number of mixture components scaled linearly with this iterative computation, limiting EM Algorithm's applicability in large-scale applications (Verbeek, 2004). The EM algorithm is easy to implement, and there is no need to set any parameters that will influence the optimization algorithm (Verbeek, 2004). However, the found solution is highly sensitive to the initial parameter values with all local optimization methods.

4.2.3 Fuzzy clustering

Fuzzy clustering is a clustering method based on the Fuzzy set developed by Zadeh (1965). The clusters are defined in fuzzy sets, with each pattern simultaneously belonging to more than one cluster, as shown in Fig. 9. Data points are assigned to two or more clusters with a degree of membership in the reflecting clusters, thereby building a non-binary relationship (Ezugwu, 2020a; Saxena et al., 2017). This way, clusters are allowed to overlap, showing what is regarded as Fuzzy overlap. Fuzzy overlap reflects the cluster boundaries' fuzziness enumerating the number of data points with significant membership in the overlapping clusters. This clustering method is beneficial for clusters of data points whose boundaries are ambiguous and not well separated (Kaufman and Rousseeuw, 1990). The degree of membership assigned to data points

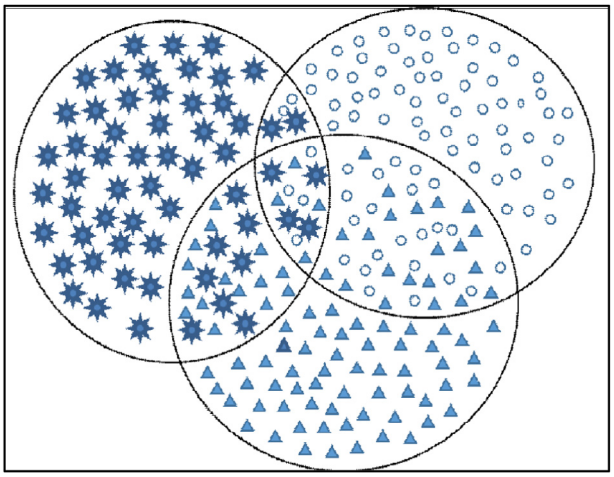


Fig. 9. Fuzzy clustering.

can also help discover the intrinsic relationship between a given object and the clusters where it is a member. Initial clustering algorithms using this method had problems with initial partition dependence, noise and outliers (Saxena et al., 2017). Subsequent algorithms based on this method tried to address these problems (Yager and Filev, 1994; Gath and Geva, 1989; Bezdek, 2000; Krishnapuram and Keller, 1993).

Fuzzy clustering can suitably handle issues such as understanding pattern's ability, mixed-media information, incomplete or noisy data, and providing faster approximate solutions (Saxena et al., 2017). For image retrieval, fuzzy clustering algorithms have been used to discover association rules and functional dependencies among data points. In line with the fuzzy clustering problems observed by Saxena et al. (2017), Gath and Geva (1989) reported three major fuzzy clustering issues: apriori knowledge of the number of clusters is not possible, there is always the need to state a cluster validity criterion before the optimal number of clusters can be determined. Also, the cluster's character and location are unknown apriori; an initial guess is needed and variability in clusters' shape, cluster densities, and the number of data points in each cluster. The time complexity of this clustering method is also high (Saxena et al., 2017). It is noteworthy that the fuzzy nature of fuzzy clusters can be removed by assigning common data to clusters with the highest degree of membership. The Fuzzy c-means is one of the most well-known algorithms based on this method (Dunn, 1973; Bezdek, 2013)

In more recent work, Shi et al. (2021) proposed FCM-RDpA combining FCM with a mini-batch gradient descent with regularization, DropRule and AdaBound (MBGD-RDA), replacing the grid partition approach in rule initialization of MBGD-RDA with fuzzy c-means to optimize a fuzzy system for a regression problem effectively. The algorithm effectively handled higher dimensional regression datasets. Borlea et al. (2021) introduced a unified form of fuzzy c-means and K-means as a single configurable algorithm. It was designed to facilitate the software implementations of FCM and K-means algorithms as one that can be configured to work as K-means or as FCM. A partitional implementation of the unified form (PIUF) was built on the UF algorithm for efficient sequential processing of large datasets and ensuring the UF algorithm's scalability in handling datasets of any size.

5 Recent work on clustering methods

Cluster validity evaluation is a major problem in the clustering algorithm. Li et al. (2020) addressed this problem by designing a cluster validity evaluation technique based on the Ratio of Deviation of Sum-of-squares and Euclid distance. The technique was evaluated on both artificial and real-world datasets, and the results show that it can dynamically obtain the near-optimal number of clusters. Chowdhury et al. (2020) introduced another technique for calculating the optimal number of clusters in a dataset. The authors designed an entropy-based initialization method, which they claim is better than other initialization methods of the K-Means algorithm. In the entropy-based initialization method, initial points are selected based on entropy and not random. The proposed technique and initialization method was evaluated on both 2-dimensional and 3-dimensional image datasets, and experiments show that it produced good results. Patil and Baidari (2019) proposed another novel method for obtaining the optimal number of clusters in a dataset, called depth difference (DeD). They used the Mahalanobis depth function to estimate the centrality of different data points within a dataset. In the study, a value between 0 and 1 is assigned to different data points in a dataset, and each value signifies the centrality or depth of each data point. The maximum depth value corresponds to the cluster centroid. The average difference between each depth value and the cluster centroid is estimated and used to obtain the optimal clusters. The optimal number of clusters is obtained by maximizing the estimated value of the depth difference.

Sinaga and Yang (2020) noted that the clustering algorithm is not mainly an unsupervised learning algorithm because it requires the number of clusters to be defined as apriori. The authors propose an unsupervised learning procedure for the K-Means clustering algorithm. The procedure can dynamically calculate the optimal number of clusters without the need for parameter selection or initialization. The method was evaluated on numerical and real-world datasets, and the results show that the proposed technique is effective. The clustering algorithm is very useful for analyzing large-scale datasets. However, selecting the optimal number of clusters for big data analysis is a major problem. Safari et al. (2020) designed a technique for automatically determining the number of clusters for large-scale datasets. They used the Bisecting K-Means algorithm and a unique splitting measure to design the technique. The technique was evaluated on several datasets, and results show the algorithm is efficient and decreases the computational costs.

Most clustering algorithm has been designed to handle datasets with one data type — either categorical or numerical datatype. However, many recent real-world applications generate datasets that contain mixed data types. Applying clustering algorithms on mixed-type datasets is difficult because of the complex relationship between the numerical and categorical attributes. Some studies proposed methods that can handle the clustering of datasets with mixed attributes. D'urso and Massari (2019) proposed a clustering-based method for handling mixed features. The model is based on the Fuzzy C-Medoids algorithm. They designed a weighting scheme to calculate each attribute's weight, representing each attribute's importance. They also calculated the dissimilarity measure for all the attribute types in the dataset. The authors combined the dissimilarity measures for each attribute and obtained a distance measure for multiple attributes. The technique was evaluated, and simulation results show that it can effectively find clusters in mixed-attributes datasets.

In another study, Behzadi et al. (2020) proposed an algorithm for clustering mixed data, called CLustering mixed-type data, including COncept Trees (ClicoT). The algorithm is based on the Minimum Description Length (MDL) principle. The algorithm leverages the natural conceptual hierarchy that exists between categorical information. The authors also designed an MDL-based objective function to integrate categorical and numerical attributes for the proposed algorithm. Experiments on synthetic and real datasets show that the algorithm is effective and robust to noise. Tran et al. (2021) proposed a framework for handling mixed-type datasets, called the COrrelation-Preserving Embedding framework (COPE). The framework uses Autoencoder Vincent et al. (2008) to learn the representations of categorical features in mixed-type data. The representations were learned by integrating two sub-networks to capture the relationship between categorical,

Table 5
Summary of recent work on clustering algorithms.

Clustering methods	Study covered	Application area	Author and year	Impact as of 2021
Ratio of Deviation of Sum-of-squares and Euclid distance	The authors designed a cluster validity evaluation technique based on Ratio of Deviation of Sum-of-squares and Euclid distance	Artificial and real-world datasets, including Iris plants, Glass dataset, Wine dataset, Gauss datasets, and shape datasets	Li et al. (2020)	5
Entropy-based initialization method for K-Means algorithm	The authors proposed a technique for calculating the optimal number of clusters in a dataset. They also designed an entropy-based initialization method for K-Means algorithm.	2-dimensional and 3-dimensional image datasets	Chowdhury et al. (2020)	5
Depth difference (DeD) for K-Means algorithm	The authors proposed a novel method for obtaining the optimal number of clusters in a dataset, called depth difference (DeD)	2-D synthetic datasets	Patil and Baidari (2019)	23
A novel technique called U-K-Means algorithm	The authors propose an unsupervised learning procedure for K-Means clustering algorithm, called U-K-Means algorithm.	Medical dataset, Iris, Seeds, Australian credit approval, Flowmeter D, Sonar, Wine, Horse, and waveform	Sinaga and Yang (2020)	23
Bisecting K-Means algorithm and a unique splitting measure	The authors designed a technique for automatically determining the number of clusters for large-scale datasets	Different large-scale datasets	Safari et al. (2020)	–
Fuzzy C–Medoids (FCMd) algorithm	The authors proposed a fuzzy clustering-based method for handling mixed features. The model is based on the Fuzzy C–Medoids algorithm. They designed a weighting scheme to calculate the weight of each attribute in a dataset.	Two simulation studies and two empirical applications were performed.	D'urso and Massari (2019)	16
Modified K-Means algorithm	The authors proposed an algorithm for clustering mixed data, called CLustering mixed-type data Including COncept Trees (ClicoT). The algorithm is based on the principle of Minimum Description Length (MDL)	Synthetic and real-world datasets including Automobile and Adult data sets.	Behzadi et al. (2020)	3-
K-Means clustering	The authors proposed a framework for handling mixed-type datasets, called COrrelation-Preserving Embedding framework (COPE). The framework uses Autoencoder Vincent et al. (2008) to learn the representations of categorical features in mixed-type data	Real-world UCI datasets including KDD99, Income, Titanic, and Echo	Tran et al. (2021)	–
k-prototype clustering algorithm	The authors introduced a mixed-type data clustering technique for risk management.	Life insurance	Yin et al. (2021)	–

numerical, and embedded data. The technique also preserves the correlation between numerical and categorical attributes. Extensive experiments were performed on different real-world datasets, and the results show that the proposed method generates very good representations of categorical features.

Death benefits are one of the largest items that affect life insurance companies. Moreover, some life insurance companies do not have a process of effectively tracking and monitoring death claims. Yin et al. (2021) introduced a mixed-type data clustering technique for risk management. They used the technique to examine the difference between actual and expected death claims. The authors used the k-prototype clustering method to extract insights from a real-world mixed-type dataset. The dataset contains policy information on life insurance. They used gap statistics to obtain optimal clusters from the dataset, and each cluster had low actual to expected death claims. The method was evaluated, and results showed that it identified a policy-holder feature, which can improve decision-making (see Table 5).

6 Discussion and open challenges

Many clustering-based algorithms have been proposed in the literature, and some of them performed remarkably well. This section presents a discussion on various issues in clustering analysis. The discussion is divided into three sub-sections. The first section presents a discussion on the performance of existing clustering algorithms. The second section presents some open issues in clustering algorithms. The third subsection presents some validation and similarity measures used in both traditional and recently proposed clustering techniques.

6.1 Performance of clustering algorithms

The performance of clustering algorithms can be categorized using nine properties (Al-Jabery et al., 2019). These properties form the major and important criteria for evaluating any clustering algorithm. The properties are presented and discussed below.

Scalability measures the running time and memory requirements for executing the clustering algorithm. It is the top priority for a clustering algorithm because of the ever-increasing data from different big data mining sources. Linear or near-linear complexity is therefore highly desirable for all clustering algorithms.

High dimensionality: This measures the algorithm's ability to handle data with many features that sometimes may be larger than the number of objects in the dataset. Identifying relevant features or capturing the intrinsic dimension is important for describing the real data structure.

Robustness: Dataset is usually not pure because there is a level of contamination introduced at the different stages of measurement, storage,

and processing, hence the need for data cleaning in data mining. It is inevitable for noise and outliers to be present in the data. The measure of the robustness of a clustering algorithm is its ability to detect and remove possible outliers and noise in the data set.

User-dependent K: Knowing the number of clusters apriori is the most fundamental problem in cluster analysis. Many existing algorithms require the number of clusters to be specified as part of the user parameters required for running the algorithm. Determining this parameter apriori is known as automatic clustering, and it has continued to attract more attention since, in many of the recent algorithms, it is difficult to decide without prior knowledge. Ability to specify the correct number of clusters assists in obtaining optimal solutions to the clustering problems in many applications. The algorithm should determine the number of clusters based on the data properties. Therefore, clustering algorithms are considered an optimization problem that can be solved using metaheuristic algorithms. Metaheuristics algorithms can be used to discover the number of clusters automatically. They can also be used to find the identified clusters.

Parameter reliance: Apart from the requirement of specifying the number of clusters apriori, there are still other algorithm-sensitive parameters that are user-defined for many of the existing clustering algorithms for users to specify for the proper functioning of such algorithms. This leaves such clustering algorithms' performances at the mercy of the wide range of users' guesses. Thus, providing practical guidance in the self-determination of such parameters by the algorithm itself or incorporating schemes that decrease reliance of algorithms on user-dependent parameters is a good measure of clustering algorithm's performance.

Irregular cluster shape: The ability to discover irregular clusters is another challenge of Clustering algorithms. There are many applications whose data may not always be formed into clusters with regular shapes. For applications with such a dataset, an optimal solution to their clustering problem will be presenting clusters of the natural shape (regular or irregular). Rather than being confined to some particular shape, a good cluster algorithm should detect irregular cluster shapes.

Order dependence: Maintaining the order of input patterns in a dataset is important in achieving correct clusters. Such characteristic is commonly in incremental or online (stream) data. Clustering algorithms for such a dataset may require that their clustering solutions vary with different orders of the presentation of the input patterns. Such algorithms must be able to achieve this. A major challenging problem in incremental learning is a reduced sensitivity or outright insensitivity to the effects of input patterns.

Visualization: Good presentation of clustering output enhances the proper interpretation of the presented result and also aids the extraction of useful information. Good visual representation of clustering output aids interpretability of the same in the problem domain, thus assisting users in understanding the results and extracting useful information from the data.

Mixed data types: Clustering algorithms are expected to be flexible enough to handle any data type in which the dataset is presented. This is very important because data obtained from different sources may be characterized by additional features, such as categorical or continuous. Some studies have shown that when operators are combined in algorithm design, it tends to poise such algorithms to robustly handle diversity in data/ population, thereby improving the quality of results within a short time (Saemi et al., 2018). Leveraging this concept, designers of clustering algorithms have the opportunity of developing algorithms capable of handling input with different attributes.

6.2 Open issues in clustering algorithms

One of the major challenges in clustering analysis is identifying the number of clusters apriori. This challenge occurs due to a lack of prior domain knowledge. It also occurs when a dataset has many dimensions with different shapes, size, density, and overlapping among groups. Although effort has been deployed towards handling this problem, it remains a major challenge. Future studies can explore nature-inspired algorithms to solve this problem (José-García and Gómez-Flores, 2016). José-García and Gómez-Flores (2016) noted that nature-inspired approaches such as bacterial foraging optimization, firefly optimization, and gravitational search algorithms could be further considered beyond non-automatic clustering to automatic clustering problems. Furthermore, the authors revealed that only a few studies thought hybrid nature-inspired algorithms in their studies. Nature-Inspired techniques can be hybridized with traditional techniques to design more efficient and faster cluster-based algorithms. Nature-inspired clustering algorithms should be hybridized through a reasonable combination of related algorithms in a fashionable and performance enhancement way that can produce improved results. Moreover, swarm intelligence-based clustering algorithms have not been fully explored in solving NP-hard problems in computational biology (Das et al., 2008b). Many more open issues in clustering analysis exist in the literature, and some of them are discussed under the sub-headings below.

Computational complexity: Some clustering algorithms have computational complexity issues, especially when applied to datasets with large instances and high-dimensional feature space. This problem can be solved by increasing computational resources production with high-capacity GPU (Shirkhorshidi et al., 2014). Moreover, exploiting parallel computing's advantage may help design patterns to deliver better clustering algorithms. Two separate studies (Shirkhorshidi et al., 2014; Zerhari et al., 2015) reported that clustering algorithms based on parallel computing appears to be very useful but suffer from the challenge of complexity as it relates to implementation. MapReduce-based clustering algorithms are an alternative to parallel computing. MapReduce-based clustering algorithms are more scalable and faster. They can considerably deploy clustering algorithms on GPU-based MapReduce frameworks to achieve better scalability and speed.

In addition to increasing computational resources with high-capacity such as GPU to tackle computational complexity issues, further refinement and enhancement to the clustering algorithm might reduce complexity. This became necessary considering each clustering algorithm's different computational complexity, yet they can achieve come commendable measure of clustering operation. We opined that finding an optimal representation of each clustering algorithm without losing its clustering operation helps minimize complexity while maintaining or maximizing the quality of clusters derived from its clustering operation.

Considering computational complexity from the general point of view of clustering categorizations, the hierarchical clustering method is known to have the complexity of $O(n^2)$, partitioning is O(n), the grid-based method is O(n), and density based method is O(n log n). Although each of these methods has a wide range of examples of clustering algorithms that often may not have the same complexity as demonstrated by the respective category they belong to, we argue that finding a minimization mechanism for representing the clustering algorithm will help reduce complexity. To further support our assertion, we show the complexity of some clustering algorithms belonging to the clustering methods (hierarchical, density-based, grid-based, and partitioned) and how they differ in their complexity compared with their parent category. The computational complexity of the clustering algorithms BIRCH is O(n), CURE is $O(n^2 \log n)$, ROCK $O(n^2 + nmmma + n^2 \log n)$, is CHAMELEON is $O(n^2)$, PAM is $O(k(n-k)^2)$, CLARA is $O(k(40 + k)^2 + k(n-k))$, CLARANS is $O(kn^2)$, DBSCAN is O(nlogn), Fuzzy C-means is O(n), K-Means is O(n), STING is O(k), WaveCluster is O(n), CLIQUE is O(Ck + mk), SOM net is $O(n^2 m)$, DENCLUE is

O(log |D|), and DBCLASD is O($3n^2$). A careful observation of each algorithm's complexity concerning the methods they fall into might reveal a slight variation, indicating that the algorithms' enhancement can reduce complexity while sustaining good and qualitative clustering results.

Refinement of Clusters: The resulting clusters from a clustering operation often require further improvement using either the same clustering algorithm or another clustery algorithm. This refinement aims to ensure that objects wrongly clustered due to inefficient similarity measures might be displaced to the cluster where they fit well. Some clustering methods, such as the divisive method, apply two approaches to the refinement task of clusters, namely monothetic and polythetic. While the former split a cluster using only one attribute, the latter split a cluster using all attributes. We considered that such approaches are already proof that more techniques can evolve to improve the quality of clusters. This refinement issue presents itself as necessary regarding the effect of wrongly classifying objects into clusters in life-threatening applications. In fact, for improved performance of the applicability of metaheuristic algorithms, such algorithms' hybrids may be considered to achieve optimal performance of the refinement task.

Speed of Convergence: A wide range of metaheuristic algorithms have inspired nature and human activities to solve optimization problems, including clustering operations effectively. Since a good convergence is one of the pointers to the effectiveness and measure of how qualitative the resulting clusters from a clustering algorithm is, continuous research into applying metaheuristic algorithms to clustering problem is encouraged. In addition to repurposing metaheuristic algorithms for solving clustering convergence problems, other related clustering operation problems that may be optimized are the sensitivity of the initialization phase and multi-objective functions involving both inter- and intra-cluster measurement and escape from local optima. Most clustering algorithms suffer from these clustering-related operations and have presented open issues allowing for further research. We argue that the effective repurposing of metaheuristic algorithms for solving these problems has great performance for clustering algorithms. For instance, variants of the firefly algorithm have been applied to the problem of initialization and escape from local optima in K-Means clustering operation (Xie et al., 2019), application of PSO to multi-objective clustering (Gong et al., 2017). We, however, note that there are other clustering operations, such as convergence, spatial clustering, and many more, which are yet to be optimized through the application of optimization algorithms.

Data dimensionality: Algorithms such as K-Means, Gaussian mixture model (GMM) clustering, maximum-margin clustering and information-theoretic clustering cannot be easily applied to problems with high-dimensional data. This problem can be solved by projecting the original data onto a low-dimensional subspace and then clustering on the feature embedding such as sparse codes (Wang et al., 2016).

Effectiveness and scalability: Effectiveness and scalability are two major challenges that open further research in clustering methods related to Big Data. The approach of deep learning has been introduced as a potential solution to this challenge. Also, decreasing the reliance of algorithms on user-dependent parameters can improve the effectiveness of clustering algorithms. Future studies can integrate domain-based requirements into a new single algorithm. Additionally, future research can develop new clustering algorithms emerging from designing solutions to some fundamental challenges of non-automatic and automatic clustering. Future research can also design improved algorithms that deal with newly occurring data without relearning from scratch.

Data object representation: Data object representation is another challenge with clustering methods. The data objects are not represented in a suitable format. Moreover, data objects are represented differently across different application areas. Some data objects are represented as feature vectors, while some are represented as graphs together with a notion of object similarity (Plant and Böhm, 2009). The differences in data object representation across different application areas portend a viable research interest. One relevant outcome of finding an efficient way for representing data in clustering operation is that it supports clustering algorithms' performance by reducing computational complexity. This allows the clustering algorithm to be scalable by identifying regions or special distribution in data. Some useful regions to identify are those aspects of the compressible data, regions that do not need to be swapped out of memory but maintained in main memory, and regions that are discardable based on noise or irrelevance to the outcome of clustering operation.

Evaluation measures: The following measures can be used as yardsticks to evaluate and compare the performance of different clustering algorithms: accuracy, algorithm stability, and dataset normalization (Chaouni et al., 2019). Moreover, there is the need to design algorithmic approaches to compare different clustering methods based on different validity indices such as internal, stability and biology indices (Bouveyron et al., 2012). Although Kokate et al. (2018) suggested that a single algorithm may not satisfy all evaluation measures, starting with one algorithmic solution may lead to further hybridized or robust solutions.

Data streams: The peculiarity of clustering makes the clustering process more demanding than clustering on static data. Kokate et al. (2018), reported some challenges of clustering methods on data streams. Clustering methods should be robust enough to deal with the existence of outliers/noise. Moreover, clustering algorithms should be capable of sharply detecting the change in context and grouping of streaming data objects to support the analysis of trends in data streams. Further, the increasing volume of data streams generated from different media like social networks should improve the computational capability and memory space optimization of clustering algorithms. More research efforts will be needed to produce adaptive models for clustering evolving data streams and improving existing context-based adaptive clustering methods.

Knowledge extraction: Another problem in clustering is knowledge extraction from big datasets. It is caused by increased data sources and generation (Ezugwu et al., 2020a). This problem poses a big challenge to the data analyst, as they cannot effectively extract knowledge from terabytes and petabytes of data. Future studies can design improved techniques that can overcome the limitation of knowledge extraction from big datasets, such as distributed clustering and parallel evolutionary algorithm. Moreover, future studies can develop new clustering methods that can choose between single-objective or multi-objective optimizations.

Therefore, in the light of the studies mentioned above and discussions, a comprehensive survey paper may be necessary which is focused on presenting an exhaustive list of clustering algorithms since it is often reported that it is hard to give a complete list of all clustering algorithms due to the diversity of information (Xu and Tian, 2015).

6.3 Cluster similarity and validation measures

This section presents the various cluster similarity and validation measures employed in traditional and recently proposed clustering techniques. A similar framework structure that briefly discussed most of the commonly used clustering validation measures was presented in Ezugwu et al. (2020a).

6.3.1 Clustering similarity measures

Cluster similarity measure gives the degree of closeness or separation of characteristics of various data points. The degree of separation or closeness can be defined explicitly or implicitly, reflecting the strength of the relationship between data points. All cluster methods clearly define cluster relationships among data objects that they are applied on, which plays a significant role in the clustering method's

success or otherwise (Patidar et al., 2012). The subsequent subsection presents a brief overview of commonly used similarity measures in traditional and recently proposed clustering techniques.

Euclidean distance: Euclidean distance is considered the standard or commonly used metric for numerical data. Simply put, it is the distance between two points, X and Y. Euclidean distance has wide acceptance in many clustering problems, and it is the default distance measure used with the K-means algorithm. The Euclidean distance is shown in Eq. (6) (Singh et al., 2013).

$$Dist_{XY} = \sqrt{\sum_{k=1}^{m} \left(X_{ik} - X_{jk}\right)^2} \tag{6}$$

where

where X and Y are two objects or sets of a certain class in cluster k.

Cosine distance: The Cosine distance measures the cosine of the angle between two data points given by Eq. (7) (Pandit and Gupta, 2011). Here θ gives the angle between two data vectors, and A, B are n-dimensional vectors and have excellent application in document similarity.

$$\theta = \frac{\arccos\left(A.B\right)}{\|A\| \, \|B\|} \tag{7}$$

Jaccard distance: The Jaccard distance or coefficient typically measures the similarity between the two data objects by evaluating the intersection divided by the data objects' unions shown in Eq. (8) (Pandit and Gupta, 2011). The Jaccard similarity measure was applied in ecological clustering species (Choi et al., 2010).

$$J\left(A, B\right) = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

Manhattan distance: Manhattan distance measures the absolute differences between coordinates of pairs of data objects X and Y, as shown in Eq. (9) (Singh et al., 2013). Manhattan distance in clustering algorithms results in hyper-rectangular-shaped clusters (Xu and Wunsch, 2005).

$$Dist_{XY} = \left|X_{ik} - X_{jk}\right| \tag{9}$$

Chebyshev distance: The determination of the absolute magnitude of the differences between coordinates of a pair of data objects X and Y is Chebyshev's goal. Eq. (10) gives the formula for Chebyshev (Singh et al., 2013).

$$Dist_{XY} = max_k \left|X_{ik} - X_{jk}\right| \tag{10}$$

Minkowski distance: Minkowski Distance is also called the generalized distance metric because given Eq. (11), when $p = 2$, the distance becomes the Euclidean distance and taking the limit when $p = \infty$ becomes Chebyshev distance metric. The Minkowski distance has a significant advantage when the embedded clusters in the dataset are compacted or isolated. Otherwise, it performs poorly (Mao and Jain, 1996).

$$Dist_{XY} = \left(\sum_{k=1}^{d} \left|X_{ik} - X_{jk}\right|^{\frac{1}{p}}\right)^{p} \tag{11}$$

Average distance: As a solution to the drawback of Euclidean distance, the average distance, which is a modified version of the Euclidean distance, was proposed to improve the results in the work of Gan et al. (2007). The average distance is defined in Eq. (12).

$$D_{ave} = \sqrt{\frac{1}{n}\sum_{k=1}^{m} \left(X_{ik} - X_{jk}\right)^2} \tag{12}$$

Weighted Euclidean distance: The Weighted Euclidean distance modifies Euclidean distance (Hand et al., 2001). It can be used when each attribute's relative importance or weights are available. This distance is defined in Eq. (13).

$$D_{weight} = \sqrt{\sum_{k=1}^{m} w_i \left(X_{ik} - X_{jk}\right)^2} \tag{13}$$

Chord distance: Given two normalized or non-normalized data points within a hypersphere of radius one, the chord length joining the points X and Y is called the chord distance. Chord distance is another variant of the Euclidean distance (Gan et al., 2007). Chord distance is defined by Eq. (14).

$$d_{chord} = \left(2 - 2\frac{\sum_{i=1}^{n} x_i y_i}{\|x\|_2 \, \|y\|_2}\right)^{\frac{1}{2}} \tag{14}$$

where $\|x\|_2$ is the L^2-norm $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

Mahalanobis distance: Mahalanobis distance is a data-driven similarity measure far from previously discussed metrics (Boriah et al., 2008). It can extract hyper ellipsoidal clusters and solve the issues caused by linear correlation among features measured (Mao and Jain, 1996) (AK, MN, & PJ., 1999). Mahalanobis distance is defined by Eq. (15) (Abonyi and Feil, 2007).

$$d_{mah} = \sqrt{(x - y)\, S^{-1} \, (x - y)^T} \tag{15}$$

where S is the covariance matrix of the dataset.

Pearson correlation: Pearson correlation has found great application in clustering gene expression data. Pearson correlation evaluates the similarities in the shape of gene expression pattern in this case (Xu and Wunsch, 2005). The Pearson correlation is defined by Eq. (16).

$$Pearson\left(x, y\right) = \frac{\sum_{i=1}^{n} \left(x_i - \mu_x\right)\left(y_i - \mu_y\right)}{\sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2} \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}} \tag{16}$$

where μ_x and μ_y are the means for $x and y$ respectively.

Multi Viewpoint-Based Similarity Measure: Multi Viewpoint-Based Similarity Measure has excellent advantages in document clustering, where multiple viewpoints can be used to make a more informative assessment of similarity. Multi Viewpoint was proposed by Sruthi and Reddy (2013), where the two data points whose similarity is to be measured must be in the same cluster, whereas the viewpoint is outside the cluster. Eq. (17) defines the multi viewpoint-based similarity measure.

$$\begin{aligned} &MVS\left(d_i, d_j | d_i, d_j \in S_r\right) \\ &= \frac{1}{n - n_r} \sum_{d_h} \cos\left(d_i - d_h, d_j - d_h\right) \left\|d_i - d_h\right\| \left\|d_j - d_h\right\| \end{aligned} \tag{17}$$

where $d_i \; and \; d_j$ are points in clusters S_r and d_h is viewpoint.

Bilateral Slope-Based Distance: A new similarity measure was proposed by Kamalzadeh et al. (2020) in their work on time-series clustering. It combines a simple representation of time series, slope of each segment of time series, Euclidean distance, and the so-called dynamic time warping. The bilateral slope-based distance (BSD) is defined as shown in Eq. (18).

$$\begin{aligned} &d_{BSD}\left(TS_i^{(1)}, TS_j^{(2)}\right) \\ &= \left|x_i^{(1)} - x_j^{(2)}\right| + \left|sin\theta_i^{(1)} - sin\theta_j^{(2)}\right| + \left|sin\theta_{i-1}^{(1)} - sin\theta_{i-1}^{(1)}\right| \end{aligned} \tag{18}$$

Distance measure for non-continuous-valued attributes: The distance measures discussed above are best applied for continuous-valued attributes. Nevertheless, not all data attributes are continuous; therefore, the distance measures are revised for categorical, binary, ordinal, or mixed type attributes.

The contingency table may be used for binary attributes, and this is given using the simple matching coefficient shown in Eq. (19).

$$\left(x_i, x_j\right) = \frac{r + s}{q + r + s + t} \tag{19}$$

For nominal attributes, the simple matching in Eq. (20) may be used.

$$d\left(x_i, x_j\right) = \frac{p-m}{p} \tag{20}$$

6.3.2 Cluster validation measure

The analogous question in cluster analysis is "how to evaluate the goodness of the resulting clusters". The resulting cluster is greatly influenced by each clustering method's parameters and initial conditions; therefore, evaluating the goodness of the clusters should consider this range of parameters and constraints. Though cluster validation is an arduous task, it plays a significant role in avoiding situations where patterns are found in the presence of noise, comparing clustering algorithms, comparing two sets of clusters, and comparing two clusters. Cluster validation criteria are usually internal or external validation (Bezdek and Pal, 1998). However, a third classification is called relative validation (Legány et al., 2006).

6.3.2.1 Internal validation criteria The underlying structure of the dataset plays a significant role in successfully partitioning the dataset. Practically, the dataset's underlying structure is usually unknown, and there is no way of knowing the correct partitioning of the dataset. The internal validation criteria measure the intra-cluster compactness and the inter-cluster separation after the dataset partitioning by the clustering algorithm. There are a variety of standards that have been proposed, which are outlined below.

Sum of squared error: Sum of squared error (SSE) is one of the most popular cluster evaluation criteria, and it is defined as follows:

$$SSE = \sum_{k=1}^{K} \sum_{\forall x_i \in C_k} \left\| x_i - \mu_k \right\|^2 \tag{21}$$

where C_k is the set of all instances in the cluster k and μ_k is the vector mean of k. The partition with the lowest SSE is considered the best (Hamilton, 1994; Tsay, 2005)

Scatter criteria: The scatter criteria (Rokach, 2005; Duda et al., 2001) is given as follows:

$$S_k = \sum_{x \in C_k} \left(x - \mu_k\right)\left(x - \mu_k\right)^T. \tag{22}$$

Condorcet's criterion: Condorcet's criterion is given as follows:

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} s\left(x_j, x_k\right) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} d\left(x_j, x_k\right) \tag{23}$$

where $s\left(x_j, x_k\right)$ and $d\left(x_j, x_k\right)$, respectively, are similarity and distance between the vectors x_j *and* x_k.

C-criterion: An extension of Condorcet's validity index is given in Fortier and Solomon (1996). The C-criterion is defined as follows:

$$\sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} \left(s\left(x_j, x_k\right) - \gamma\right) + \sum_{C_i \in C} \sum_{x_j \in C_i; x_k \notin C_i} \left(\gamma - s\left(x_j, x_k\right)\right) \tag{24}$$

where γ is a threshold value.

Category utility metric: Given a set of entities, the binary feature set of size n is defined as

$$F = \left\{f_i\right\}, i = 1, 2, \ldots, n$$

and the binary category $C = \left\{c, \overline{c}\right\}$ is defined as follows:

$$CU\left(C, F\right) = \left[p\left(c\right) \sum_{i=1}^{n} p\left(f_i|c\right) \log p\left(f_i|c\right) + p\left(\overline{c}\right) \sum_{i=1}^{n} p\left(f_i|\overline{c}\right) \log p\left(f_i|\overline{c}\right)\right] - \sum_{i=1}^{n} p\left(f_i\right) \log p\left(f_i\right), \tag{25}$$

given that: $p(c)$ is the prior probability of an entity belonging to the positive category c $p\left(f_i|c\right)$ is the conditional probability of the feature f_i given that it belongs to the positive category c $p\left(f_i|\overline{c}\right)$ is the conditional probability of the feature f_i given that it belongs to the positive category $\overline{c}$ and $p\left(f_i\right)$ is the previous probability of the entity (Corter and Gluck, 1992; Ezugwu et al., 2020a).

Bayesian information criterion (BIC) index: The problem of overfitting the partitions generated by the clustering algorithm is huge, and BIC tries to solve it. BIC is a minimization problem (Raftery, 1986) and is defined as follows:

$$BIC = -\ln\left(L\right) + v \ln\left(n\right) \tag{26}$$

where n is the number of entities, L is the likelihood of the parameters to generate the data in the model, and v is the number of free parameters in the Gaussian model.

Calinski–Harabasz index: The Calinski–Harabasz validity index measures the compactness or closeness of the clusters by calculating the distances between the points in a cluster to their centroids. Likewise, the separation is calculated by measuring the distance from the centroids to the global centroid (Calinski and Harabasz, 1974; Ezugwu et al., 2020a). This index is defined as

$$CH = \frac{\text{trace}\left(\text{SB}\right)}{\text{trace}\left(\text{Sw}\right)} \cdot \frac{n_p - 1}{n_p - k} \tag{27}$$

where (SB) is the inter-cluster scatter matrix, (Sw) the intra-cluster scatter matrix, np is the number of entities in a cluster, and k the number of clusters.

Davies–Bouldin index (DB): The DB index evaluates the average inter-cluster similarity between any two clusters and their nearest. This information is critical to the success of the DB index (Davies and Bouldin, 1979). For better results, DB is minimized. The Davies–Bouldin index is defined as follows:

$$BD = \frac{1}{c} \sum_{i=1}^{c} \underset{i \neq j}{Max} \left\{ \frac{d\left(x_i\right) + d\left(x_j\right)}{d\left(c_i, c_j\right)} \right\} \tag{28}$$

where c is the number of clusters, i, j are cluster labels, $d\left(x_i\right)$ and $d\left(x_j\right)$ are all entities in clusters i and j, $d\left(c_i, c_j\right)$ is the distance between the cluster centroids.

Silhouette index: The silhouette index requires that information about the compactness and separation of at least two clusters be known (Rousseeuw, 1987).

Given a cluster, X_j $(j = 1, ..c)$, the index assigns to the ith entity of X_j the silhouette width, $s(i) = (i = 1, \ldots, m)$. This value gives a degree of likelihood of the ith sample belonging in the cluster X_j. The index is defined as:

$$s\left(i\right) = \frac{\left(b\left(i\right) - a\left(i\right)\right)}{Max\left\{a\left(i\right), b\left(i\right)\right\}} \tag{29}$$

where $a(i)$ is the average distance between the ith entity in the cluster and the remaining entities of cluster X_j; $b(i)$ is the minimum average distance between the ith s and all of the entities clustered in X_k $(k = 1, \ldots c; k \neq j)$.

Dunn index: The Dunn index basically looks for the ratio between the smallest inter-cluster distance and the largest intra-cluster distance in a partitioning (Dunn, 1973). The Dunn index is defined as follows:

$$Dunn = min_{1 \leq i \leq c} \left\{ \min \left\{ \frac{d\left(c_i, c_j\right)}{\underset{1 \leq k \leq c}{max}\, d\left(X_k\right)} \right\} \right\} \tag{30}$$

where $d\left(c_i, c_j\right)$ is the distance between cluster X_i and X_j; $d\left(X_k\right)$ represents the distance between members of cluster (X_k) and c is the number of clusters in the dataset. The Dunn index is a maximization problem with setbacks that include its time complexity, and it is being affected by noise in datasets.

NIVA index: The NIVA validation index (Rendon et al., 2008) is defined as follows:

$$NIVA\left(C\right) = \frac{Compac\left(C\right)}{SepxG\left(C\right)} \tag{31}$$

where $Compac\,(C)$ is the average compactness of the cluster C and $SepxG\,(C)$ is the average separability of cluster C.

Gamma Index: The gamma index (Baker and Hubert, 1975) is defined as:

$$G\,(C) = \frac{\sum_{c_k \in C} \sum_{x_i, x_j \in c_k} dl\,(x_i, x_j)}{n_w \left(\binom{N}{2} - n_w\right)} \tag{32}$$

where $dl\,(x_i, x_j)$ is the number of all object pairs in X.

Score function: The scoring index evaluates the dispersion amongst clusters by estimating the distance between cluster centroids and the global centroid. The score function measures the degree of closeness of the clusters by estimating the distance from the points in a cluster to their centroid (Saitta and Smith, 2007; Saitta, Raphael, and Smith, 2007; Ezugwu et al., 2020a). The index is defined as follows:

$$SF\,(C) = 1 - \frac{1}{e^{ebcd(C)+wcd(C)}} \tag{33}$$

where

$$bcd\,(C) = \frac{\sum_{c_k \in C} |c_k|\, d_e\overline{(c_k, X)}}{N \times K}$$

and

$$wcd\,(C) = \sum_{c_k \in C} \frac{1}{|c_k|} \sum_{x_i \in c_k} d_e\overline{x_i, (c_k)}.$$

C-Index: The C-index (Dalrymple-Alford, 1970) is defined as:

$$CI\,(C) = \frac{S\,(C) - S_{\min(C)}}{S_{\max(C)} - S_{\min(C)}} \tag{34}$$

where

$$S\,(C) = \sum_{C_k \in C} \sum_{x_i, x_j \in C_k} d_e\,(x_i x_j)$$

$$S_{\min(C)} = \sum \min\,(n_w)_{x_i, x_j \in X} \{d_e\,(x_i x_j)\}$$
$$S_{\max(C)} = \sum \max\,(n_w)_{x_i, x_j \in X} \{d_e\,(x_i x_j)\}$$

Sym-index: The sym-index (Bandyopadhyay and Saha, 2008). It is defined as follows:

$$Sym\,(C) = \frac{max_{C_k, C_l \in} \left\{d_e\overline{(c_k, c_l)}\right\}}{K \sum_{c_k \in} \sum_{x_i \in c_k} d^*_{ps}\,(x_i, c_k)} \tag{35}$$

COP Index: The COP index is measured as the distance from the centroids to the cluster points. Likewise, the separation is measured as the largest distance between neighbors (Arbelaitz et al., 2013). It is defined as follows:

$$COP\,(C) = \frac{1}{N} \sum_{c_k \in} |c_k| \frac{\frac{1}{|c_k| \sum_{x_i \in c_k} d_e\overline{(x_i c_k)}}}{min_{x_i \notin c_k} max_{x_i \in c_k} d_e\,(x_i x_j)} \tag{36}$$

Negentropy Increment: Measuring the normality of clusters instead of the compactness or separation of the clusters is the goal of the negentropy index (Lago-Fernández and Corbacho, 2010)

$$NI\,(C) = \frac{1}{2} \sum_{c_k \in C} p\,(c_k) \log|\sum_{c_k}| - 1/2 \log|\sum_{X}| - \sum_{c_k \in C} p\,(c_k)\, logp(c_k) \tag{37}$$

SV-Index: The SV-index evaluates cluster separation as a measure between nearest neighbors. The compactness measures border points to centroids of the cluster (Žalik and Žalik, 2011). It is defined as follows:

$$SV\,(C) = \frac{\sum_{c_k \in C} min_{c_l \in C \setminus c_k} \{d_e\overline{(c_k, c_l)}\}}{\sum_{c_k \in C} 10/|c_k| \sum max_{x_i \in c_k} \left(0.1\,|c_k|\right) \{d_e\,(x_i \overline{c_k})\}} \tag{38}$$

OS-Index: The OS-index (Drewes, 2005) is defined as follows:

$$OS\,(C) = \frac{\sum_{c_k \in C} \sum_{x_i \in c_k} OV(x_i, c_k)}{\sum_{c_k \in C} 10/|c_k| \sum max_{x_i \in c_k} \left(0.1\,|c_k|\right) \{d_e\,(x_i \overline{c_k})\}} \tag{39}$$

The modified Hubert Γ statistic: The modified Hubert Γ statistic (Theodoridis and Koutroubas, 1999) is defined as follows:

$$\Gamma = \left(\frac{1}{M}\right) \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P\,(i, j)\,.Q\,(i, j) \tag{40}$$

where the dimension of the dataset is N, $M = \frac{N(N-1)}{2}$, P is the proximity matrix of the dataset and Q is an $N \times N$ matrix.

SD validity index: The SD validity index measures the mean intra- and inter-cluster scattering (Halkidi et al., 2002). The definition for this index is as follows:

$$SD\,(n_c) = a.Scat\,(n_c) + Dis\,(n_c) \tag{41}$$

where

$$Scat\,(n_c) = \frac{\frac{1}{n_c} \sum_{i=1}^{n_c} \left\|\sigma\,(v_i)\right\|}{\|\sigma\,(X)\|}$$

$$Dis\,(n_c) = \frac{D_{max}}{D_{min}} \sum_{k=1}^{n_c} \left(\sum_{z=1}^{n_c} \|v_k - v_z\|\right)^{-1}$$

Dbw validity index: The clusters' underlying characteristics are used by S_Dbw validity index aims to measure the validity of the results from the clustering algorithm (Halkidi and Vazirgiannis, 2001). It is defined as follows

$$S_{Dbw(n_c)} = Scat\,(n_c) + Dens_{bw(n_c)} \tag{42}$$

where

$$Dens\_bw(n_c) = \frac{1}{n_c.(n_c - 1)} \sum_{i=1}^{n_n} \left(\sum_{\substack{j=1 \\ i \neq j}}^{n_c} \frac{density(u_{ij})}{\max\{density\,(v_i), density\,(v_j)\}}\right)$$

where v_i, v_j are the centroids of cluster $c_i c_j$, and u_{ij} the middle point of the line segment.

Root-mean-square standard deviation (RMSSTD): The RMSSTD evaluates the square root of the variance of all the attributes used in the clustering process (Davies and Bouldin, 1979). It is defined as:

$$RMSSTD = \left[\frac{\sum_{\substack{i=1\ldots nc \\ j=1\ldots v}} \sum_{k=1}^{n_{ij}} \left(x_k - \overline{x_k}\right)^2}{\sum_{\substack{i=1\ldots nc \\ j=1\ldots v}} \left(n_{ij} - 1\right)}\right] \tag{43}$$

R-squared (RS): The RS index (Sharma, 1996) is defined as follows:

$$RS = \frac{SS_b}{SS_t} = \frac{SS_t - SS_w}{SS_t}. \tag{44}$$

Compact-Separated (CS) index: The CS index measures the ratio of the sum of within-cluster scatter to between-cluster separation (Kosters and Laros, 2007). Minimizing the CS index leads to better clustering. Let the within-cluster scatter be denoted as X_i and the between-cluster separation be represented as X_j, such that the distance measure V is given as $V\,(X_i, X_j)$. Hence, the CS index for a clustering Q is computed as follows (Ezugwu et al., 2020a).

$$CS\,(Q, V) = \frac{\frac{1}{P} \sum_{i=1}^{P} [\frac{1}{D_n} \sum_{X_i \in Q_i} max_{X_j \in Q_i} \{V(X_i, X_j)\}]}{\frac{1}{P} \sum_{i=1}^{P} \left[min_{j \in P, j \neq i} \left\{V\,(x_i, x_j)\right\}\right]} = \frac{\sum_{i=1}^{P} \left[\frac{1}{Q_i} \sum_{X_i \in Q} max_{X_j \in Q} \{V\,(X_i, X_j)\}\right]}{\sum_{i=1}^{P} [min_{j \in P, j \neq i} \left\{V\,(x_i, x_j)\right\}]} \tag{45}$$

where $|D_n|$ represents the number of data points in cluster P, the function $V\,(X_i, X_j)$ is the distance between within-cluster scatter X_i

and between-cluster separation X_j, $V\left(x_i, x_j\right)$ is the distance of data points d from their centroids, and P is the number of clusters in Q.

Ball–Hall index: The Ball–Hall index gives the mean of the mean dispersion of all the clusters, and it is given as follows (Ball and Hall, 1965):

$$Ball\text{–}Hall\ index\,(C) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i\in I_k}\left\|M_i^{\{k\}} - G^{\{k\}}\right\|^2 \tag{46}$$

Banfeld–Raftery index: The Banfeld–Raftery index evaluates the weighted sum of the logarithms of the traces of the variance of the covariance matrix of each cluster. The definition is given as follow (Banfield and Raftery, 1993):

$$C = \sum_{k=1}^{K} n_k log\left(\frac{Tr(WG)^{\{k\}}}{n_k}\right) \tag{47}$$

Det Ratio index: The Det Ratio index is defined as follows (Scott and Symons, 1971)

$$Det\ Ratio = \frac{\det(T)}{\det(WG)} \tag{48}$$

where T is the total scatter matrix and WG is the individual matrices

Baker-Hubert Gamma index: Given two vectors, A and B, with the same dataset size, the Baker-Hubert Gamma index evaluates the two vectors' correlation. The index is an adaptation of the Γ index, and it is defined as follows (Baker and Hubert, 1975):

$$C = \Gamma = \frac{S^+ - S^-}{S^+ + S^-} \tag{49}$$

where $S^+ = \sum_{(r,s)\epsilon I_B}\sum_{(u,v)\epsilon I_W} 1_{\{d_{uv}<d_{rs}\}}$ and $S^- = \sum_{(r,s)\epsilon I_B}\sum_{(u,v)\epsilon I_W} 1_{\{d_{uv}>d_{rs}\}}$

GDI index: The Generalized Dunn's Index (GDI) evaluates the intra-cluster and inter-cluster distances (Bezdek and Pal, 1998)

$$C = \frac{min_{k\neq k'}\delta\left(C_k, C_{k'}\right)}{max_k\Delta\left(C_k\right)} \tag{50}$$

where δ is a measure of inter-cluster distance and Δ is a measure of intra-cluster distance, with $1 \le k \le K and 1 \le k' \le K$

G-plus index: The G-plus index (Rohlf, 1974) is defined as follows:

$$G+ = \frac{2S^-}{N_T(N_T-1)} \tag{51}$$

Ksq_DetW index: Also denoted as $k^2|W|$, this index (Marriot, 1975) is defined as follows:

$$C = K^2\det(WG) \tag{52}$$

where WG is the individual cluster matrices

Log_Det_Ratio index: This index is the logarithmic version of Det_Ratio given in Eq. (53), and it is defined as follows (Scott and Symons, 1971):

$$C = Nlog\left(\frac{\det(T)}{\det(WG)}\right) \tag{53}$$

Log_SS_Ratio index: The Log_SS_Ratio index evaluates the ratio of the traces of matrices BG and WG. The definition of the index is given as follows (Hartigan, 1975):

$$C = log\left(\frac{BGSS}{WGSS}\right) \tag{54}$$

McClain–Rao index: The McClain–Rao index calculates the mean ratio of intra-cluster and inter-cluster distances (McClain and Rao, 1975). The index is defined as follows:

$$C = \frac{N_B S_W}{N_W S_B} \tag{55}$$

where

$$S_W = \sum_{k=1}^{K}\sum_{\substack{i,j\in I_k\\ i<j}} d\left(M_i, M_j\right)$$

$$S_B = \sum_{k<k'}\sum_{\substack{i\in I_k, j\in I_{k'}\\ i<j}} d\left(M_i, M_j\right)$$

N_B is the total number of distances between pairs of points that do not belong to the same clusters and N_W is the total number of distances between pairs of points that belong to the same clusters

PMB index: The PBM index evaluates the distance between the points and their barycenters. Furthermore, it also evaluates the distances between the barycenters Pakhira et al. (2004). PMB is the acronym for the author's initials of the names of its authors (Pakhira, Bandyopadhyay, and Maulik)

$$C = \left(\frac{1}{K}\times\frac{E_T}{E_W}\times D_B\right)^2 \tag{56}$$

where

$$D_B = \max_{k<k'} d\left(G^{\{k\}}, G^{\{k'\}}\right)$$

$$E_W = \sum_{k=1}^{K}\sum_{i\in I_k} d\left(M_i, G^{\{k\}}\right)$$

$$E_T = \sum_{i=1}^{N} d(M_i, G)$$

Point-Biserial index: The Point-Biserial index (Milligan, 1981) is defined as follows:

$$C = s_n \times r_{pb}(A,B) = \left(\frac{S_W}{N_W} - \frac{S_B}{N_B}\right)\frac{\sqrt{N_W N_B}}{N_T} \tag{57}$$

where

$$r_{pb}(A,B) = \frac{M_{A_1} - M_{A_0}}{s_n}\sqrt{\frac{n_{A_0} n_{A_1}}{n^2}}$$

M_{A_1} is the mean of the intra-cluster distances and M_{A_0} is the mean of the inter-cluster distances. s_n is the standard deviation of A, n_{A_0}, n_{A_1} are the number of elements in each group. Set A represents distances between pairs of cluster points, and the value of B is 1 if a pair of points are in the same cluster and 0 otherwise.

Ratkowsky–Lance index: The Ratkowsky–Lance index (Ratkowsky and Lance, 1978). is defined as follows:

$$C = \sqrt{\frac{\overline{R}}{K}} = \frac{\overline{c}}{\sqrt{K}} \tag{58}$$

where

$$\overline{c}^2 = \overline{R} = \frac{1}{p}\sum_{j=1}^{p}\frac{BGSS_j}{TSS_j}$$

$BGSS_j$ is the jth diagonal term of the matrix BG

Ray–Turi index: The Ray–Turi index (Ray and Turi, 1999) can be defined as follows

$$C = \frac{1}{N}\frac{WGSS}{\min_{k<k'}\Delta_{kk'}^2} \tag{59}$$

The numerator is the mean of the squared distances of all points concerning the barycenter of the cluster they belong. The denominator is the minimum of the squared distances between all the cluster barycenters

Scott–Symons index: The Scott–Symons index evaluates each cluster, the weighted sum of the logarithms of the variance–covariance matrix's

determinant (Scott and Symons, 1971).

$$C = \sum_{k=1}^{K} n_k \log det\left(\frac{WG^{\{k\}}}{n_k}\right) \tag{60}$$

where
$WG^{\{k\}}$ are matrices and the determinants of the matrices are positive.

Tau index: The Tau index can be defined as follows:

$$C = \frac{s^+ - s^-}{\sqrt{N_B N_W \left(\frac{N_T(N_T-1)}{2}\right)}} \tag{61}$$

when inter-cluster and intra-cluster distances are equal, the numerator is not affected because $s^+ and s^-$ do not count ties

Trace_W index: This index is defined as follows (Edwards and Cavalli-Sforza, 1965):

$$C = Tr(WG) = WGSS \tag{62}$$

where WGSS is the within-cluster sum of squares and WG is the sum of all the clusters

Trace_WiB index: The Trace_WiB index (Friedman and Rubin, 1967) can be defined as follows

$$C = Tr(WG^{-1}.BG) \tag{63}$$

Wemmert–Gançarski index: The weighted mean of all quantities (J_k) for all the clusters is evaluated by the Wemmert–Gançarski index. This index is defined as follows

$$C - \frac{1}{N}\sum_{k=1}^{K} \max\{0, n_k - \sum_{i \in I_k} R\left(M_i\right)\} \tag{64}$$

where for M belonging to cluster C_k

$$J_k = \max\{0, 1 - \frac{1}{n_k}\sum_{i \in I_k} R\left(M_i\right)\}$$

$$R(M) = \frac{\left\|M - G^{\{k\}}\right\|}{\min_{k' \neq k}\left\|M - G^{\{k'\}}\right\|}$$

Xie–Beni index: This index gives the ratio between the mean quadratic error and the minimum of the squared distances between the clusters' points (Xie and Beni, 1991).

$$C = \frac{1}{N}\frac{WGSS}{\min_{k<k'} \delta_1\left(C_k, C_{k'}\right)^2} \tag{65}$$

$$\delta_1\left(C_k, C_{k'}\right) = \min_{\substack{i \in I_k \\ j \in I_{k'}}} d(M_i, M_j)$$

6.3.2.2 External quality criteria measures External quality criteria require that knowledge about the dataset, particularly the underlying structure and number of clusters embedded in the dataset, is known. This knowledge is always not available in real-world scenarios. The basic idea is to match the result of the partition with the predefined structure of the dataset.

Mutual Information-Based Measure: The goal of the mutual information-based measure is to evaluate the mutual interdependence between the partition result and the dataset's underlying structure (Strehl and Ghosh, 2000).

For m instances in clusters $C = \{C_1, C_2, \ldots, C_g\}$ and target attribute z with domain $dom(z) = \{c_1, c_2, \ldots, c_k\}$, the index is defined as follows:

$$C = \frac{2}{m}\sum_{i=1}^{g}\sum_{h=1}^{k} m_{i,h} log_{g,k}\left(\frac{m_{i,h}.m}{m_{.,h}.m_{i,.}}\right) \tag{66}$$

where $m_{i,h}$ is the number of instances in cluster C_i and in class $c_h, m_{.,h}$ denotes the total number of instances in the class c_h, and $m_{l,.}$ denotes the number of instances in cluster C_i.

Rand Index: Rand index refers to the similarities between partitions of the clustering algorithm and the dataset's underlying structure (Rand, 1971; Ezugwu et al., 2020a). The index is defined as follows:

$$RAND = \frac{TP + TN}{TP + FP + FN + TN} \tag{67}$$

F-measure: The consequence of equal weighting for the false positives and false negatives is that we usually end up with undesirable features. The F-measure index uses the weighting recall parameter $\eta > 0$ to balance the false negatives (Rijsbergen, 1979; Ezugwu et al., 2020a). The F-measure is defined as follows:

$$F = \frac{\left(\eta^2 + 1\right).P.R}{\eta^2.P + R} \tag{68}$$

where P is the precision rate and R is the recall rate.

The range of recall starts from no effect ($\eta = 0$) to more effect as η increases (indicating a higher clustering quality).

Jaccard Index: The Jaccard index evaluates the ratio of the intersection of both dataset elements and the union of the elements in both datasets. The Jaccard index is defined as follows:

$$JJ\left(AB\right) = \frac{A \cap B}{A \cup B} = \frac{TP}{TP + FP + FN} \tag{69}$$

if A and B are empty, then $0 \leq J\left(AB\right) \leq 1$.

Fowlkes–Mallows Index: Fowlkes–Mallows index measures the compactness of clusters obtained from a clustering algorithm, maximizing the index results in higher similarities (Fowlkes and Mallows, 2010; Ezugwu et al., 2020a). The Fowlkes–Mallows index is defined as follows

$$FM = \sqrt{\frac{TP}{TP + FP}.\frac{TP}{TP + FN}} \tag{70}$$

NMI measure: The normalized mutual information (NMI) is defined as follows:

$$NMI\left(X, Y\right) = \frac{I\left(X, Y\right)}{\sqrt{H\left(X\right)H\left(Y\right)}} \tag{71}$$

where, $I(X, Y)$ is the mutual information between two random variables X and Y and $H(X)$ denotes the entropy of X, X is the partition by a clustering algorithm, and Y represents the true labels of the dataset (Legány et al., 2006).

Purity: The purity for each cluster P_j is defined as:

$$P_j = \frac{1}{n_j} Max_i\left(n_j^i\right) \tag{72}$$

The purity for the set of clusters is calculated as a weighted sum of the individual purities (Kovács and Ivancsy, 2006). This is given as:

$$Purity = \sum_{j=1}^{m} \frac{n_j}{n} P_j \tag{73}$$

where n_j denote the size of cluster j, m is the number of clusters, and n is the total number of entities.

Entropy: Entropy increases as the classification of objects in a cluster becomes more varied. If all the objects in the cluster belong to one label, then the entropy is 0 (Drewes, 2005; Ezugwu et al., 2020a). In this case, entropy is defined as follows:

$$E_j = \sum_i p_{ij} \log\left(p_{ij}\right) \tag{74}$$

Relative validation: This scheme aims to validate the partitions of a clustering algorithm by specifying the best clustering scheme possible under certain assumptions and parameters. The relative validation

involves a lot of statistical testing (Halkidi et al., 2002; Ezugwu et al., 2020a). Assuming the clustering problem is defined as thus:

" Let P_{alg} be the set of parameters associated with a specific clustering algorithm (e.g. the number of clusters n_c). Among the clustering schemes C_i, $i = 1, \ldots, n_c$, defined by a specific algorithm, for different values of the parameters in P_{alg}, choose the one that best fits the data set". (Halkidi et al., 2002; Ezugwu et al., 2020a).

The following cases hold:

- P_{alg} does not contain n_c as a parameter.

The idea here is to tune the parameter over a wide range of values and run the clustering algorithm, then choose the maximum range for which n_c is constant. Normally, $n_c \ll N$ where N is the number of tuples

- P_{alg} contains n_c as a parameter.

Define a maximum and minimum range first, then run the algorithm r times over each n_c between the minimum and maximum range, tuning the parameters during each run. Then plot the best values of the index obtained against λ. The plot may indicate the best cluster.

7 Trending application areas of clustering algorithms

Clustering algorithms can be applied to different domains. This section provides diverse areas that cluster analysis that has been successfully utilized. Specifically, we present the applicability of the clustering algorithms reviewed in Section 4 to the field of medicine, financial sector, artificial intelligence, aviation sector, marketing and sales sector, industries and manufacturing context, urban development, privacy protection, and robotics. Fig. 10 summarizes these fields by presenting a taxonomy of the applicability of clustering algorithms.

7.1 Web usage

The ever-growing electronic data size originating from web applications' proliferation has motivated the exploration of hidden information from text content. Nirkhi and Hande (2008) presented a summary on the use of web page rank algorithms (such as Hyperlink Induced Topic Search, page rank) and web page-based clustering algorithms (such as Suffix Tree, Vivisimo and Lingo clustering algorithms) (Nirkhi and Hande, 2008). Similarly, Ivancsy and Kovacs (2006) listed some newer and relevant approaches to web clustering algorithms, such as Fuzzy clustering.

The corresponding web server logs information for each user is accessing a page, including IP address, time of access, file path, browser, and amount of transferred data. Vast volumes of web server log data are generated every day, and they can be used for commercial and non-commercial applications such as designing online shops or providing users with personalized content in digital libraries. Madhulatha (2015) supported this claim that clustering algorithms have been used for library book ordering, document classification, clustering web log data to discover groups of similar access patterns. Another study proposed using the KEA-Means algorithm, which combines the keyphrase extraction algorithm and the K-Means algorithm to generate the number of web documents clusters from a dataset (Ware and Dhawas, 2012). Lin et al. applied a novel algorithm named hierarchical clustering (HSClus) to extract a similarity matrix among pages via in-page and cross-page link structures. This resulted in clusters that hierarchically groups densely linked web pages into semantic clusters (Lin et al., 2009). Tang Rui et al. (2012) carried out an investigative study to discover the possibilities of applying other nature-inspired-based optimization algorithms: Fireflies, Cuckoos, Bats, and Wolves for performing clustering over Web Intelligence data. Sardar and Ansari (2018a) applied the K-Means algorithm using the MapReduce programming model to the task of document clustering (Sardar and Ansari, 2018a,b).

7.2 Speech processing

Speech is an important communication medium among humans and even animals as well. Large data sourced from communication on phone and tele-media platforms may require clustering methods to discover knowledge. Sonkamble and Doye proposed a clustering algorithm named Modified K-MeansLBG. They applied it to obtain a good codebook for vector quantization which is usually used in speech coding, image coding, speech recognition, speech synthesis and speaker recognition (Sonkamble and Doye, 2012). Rovetta et al. applied fuzzy clustering to the task of recognizing emotion from speech signals in addition to the use of probabilistic, possibilistic and graded-possibilistic c-means techniques (Rovetta et al., 2019). Neel (2005) used K-Means and fuzzy K-Means clustering to capture phonetic classification effectively. They used clustering to identify the kinds of phonetic features to the group and discover automatic clustering efficiency in clustering speech data. In another study, Bach and Jordan (2006) applied spectral clustering to the blind one-microphone speech separation problem. They cast the problem as one of the segmentation of the spectrogram. Vani and Anusuya (2019) investigated the performances of clustering techniques (like K-Means, Fuzzy C means, and Kernel Fuzzy C Means algorithms) for clustering noisy speech signals. Single-space and Multi-level clustering methods were applied to the speech processing system's problem (Räsänen, 2007). Fig. 10 gives the taxonomy of clustering algorithm applications.

7.3 Medical science: Disease onset and progression

The Healthcare sector is one of the predominant fields of human endeavor that requires improvement alongside contemporary society's development. It is one of the main sectors that generally impact the members of the public. Presently, clustering analysis has solely depended on the transformation of healthcare services. Clustering algorithm has tremendously contributed to disease diagnosis aspects of healthcare services such as simplification of the ocular disease detection process, utilizing retinal blood vessels segmentation (Waheed et al., 2015), detection of tumors using K-Means clustering algorithm (Patel et al., 2013), and detection of neovascularization in retina image by employing multivariate m-Medoids clustering algorithms (Akram et al., 2013). Besides the diagnosis, medical imaging is another crucial aspect of medicine that plays a role in inpatient treatment. It assists the medical research domain in investigating different parts of human anatomy and understanding the effects of particular illnesses. For instance, mean shift clustering has been employed in blood oxygen level-dependent function MRI activation detection (Ai et al., 2014). Also, semi-supervised clustering has been utilized for brain image segmentation (Saha et al., 2016). Similarly, fuzzy clustering – a hybrid approach – has been successfully applied to segment inhomogeneous medical images (Rastgarpour et al., 2014). Medical image analysis uses spectral clustering (Kuo et al., 2014). Recent research has been conducted on genetics and hype found in modern medicine to produce a personalized treatment system. Thus, clustering analysis has facilitated such a research process (Datta and Datta, 2003; Aouf et al., 2008; Oyelade et al., 2016).

In a study (Magoeva et al., 2018), authors investigated the possibility of applying some clustering algorithms (such as DBSCAN and K-Means) to detect critical patient patients conditions with coronary syndrome using medical parameter time series. The study successfully performed a preliminary analysis showing that outlier clustering is viable for potential outlier classification and analysis. Their results revealed that the investigated clustering algorithms achieve a moderate performance in critical patient detection. Another study (Newcomer et al., 2011) reported using cluster analysis to identify sub-populations of complex patients who may benefit from targeted care management strategies. This was achieved by using a massive 2-year cohort of health maintenance organization members with 2 or more chronic conditions.

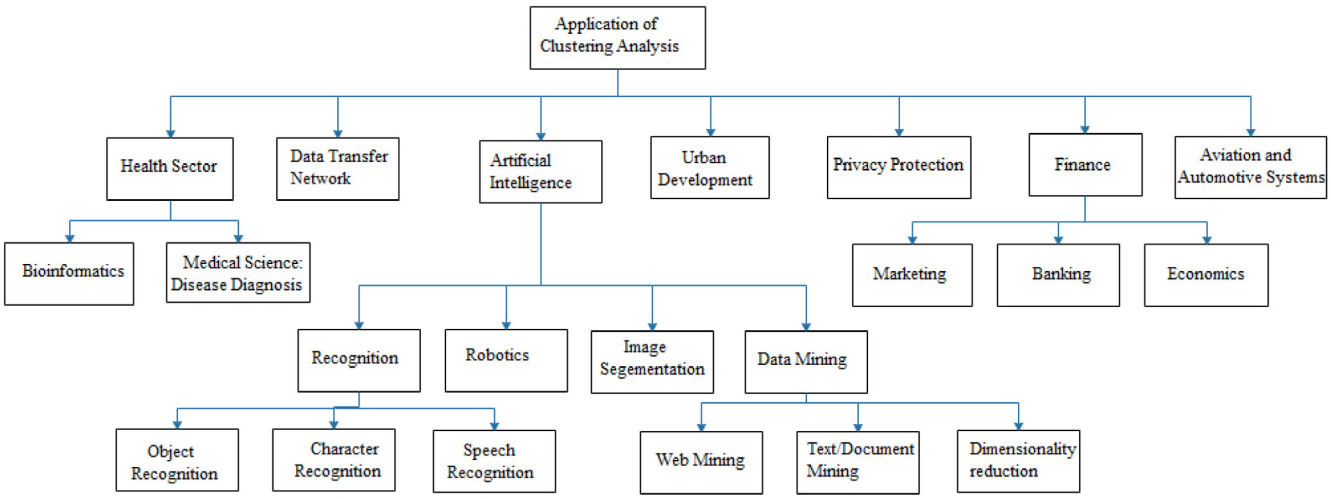


Fig. 10. Taxonomy of clustering algorithm application.

A new study proposes a medical-oriented big data clustering algorithm using an improved immune evolutionary method (evolutionary algorithm and FCM) by applying the clustering algorithm to medical data, including encoding, constructing fitness function, and selecting genetic operators (Yu et al., 2020). To unravel patient heterogeneity, Forte et al. (2019) applied some selected hard and soft clustering algorithms (based on the expected overlap between sub-phenotypes and the size of the dataset) to conduct a panorama of clustering analyzes using heterogeneous and complex ICU data. In a different study (Venkataramana et al., 2017), the authors attempted to apply the clustering method (Fuzzy C-Means and K-Means) to speed up the process of analyzing patients' samples in high volume hospitals to aid decide on the stage of the disease.

7.4 Image processing and segmentation

Clustering methods have been applied to the problem of image segmentation and image processing. While image processing describes a situation where a wide range of likely computational operations is (are) applied to an image for knowledge discovery of image refinement, a component of image processing is image segmentation. Image segmentation can be expressed as exhaustive partitioning of an input image into regions so that each is considered homogeneous concerning some image property. Chopade and Sheetlani (2017) reported that studies had employed evolutionary fuzzy clustering methods with knowledge-based evaluation to handle image segmentation problems. The approach of the GAs clustering method was applied to the problem, which was formulated as an optimization so that clustering of small regions in color feature space was achieved. A study by Saxena et al. (2017) reported that K-Means had been successfully applied to image segmentation. The quality of images presented by Magnetic Resonance Imaging (MRI) to visualize objects and living organisms' internal structures made it a candidate for the segmentation task. This task can be formulated as a clustering problem from MRI images so that feature vectors obtained through transformation image measurements and pixel positions are grouped into several structures. Dhanachandra et *al.* combined K-Means and subtractive cluster algorithms for image segmentation. They achieved this by pre-processing the image using partial stretching enhancement and the hybridized algorithm to generate the initial centers.

The K-Means algorithm then uses the initial centers to segment the image before removing unwanted regions using a medial filter (Dhanachandra et al., 2015). In a similar work (Chena et al., 2015), the authors applied the use of the DP clustering algorithm. They directly provided the cluster number of the image based on the decision graph and identified cluster centers to carry out hierarchical segmentation. In another study (Nameirakpam and Jina, 2015), the authors combined local histogram equalization and K-Means clustering for image segmentation. Bora and Gupta carried out a similar study using a hard K-Means clustering algorithm with cosine distance measure while filtering and analyzing the segmented image with the Sobel filter and watershed algorithm (Bora and Gupta, 2014).

Gulhane et al. (2012) investigated the association that exists between the density of data points in the given data set of pixels of an image using K-Means and M-step differently. Parida (2018) carried out image segmentation using fuzzy C-means clustering by applying it to the variance feature image to separate its transitional features (Parida, 2018). An attempt to apply clustering algorithms to the novel Covid-19 disease was made by the authors in ElazizID et al. (2020). They carried out image segmentation on COVID-19 Computed Tomography (CT) images. They proposed a clustering method to improve the density peaks clustering (DPC) combined with generalized extreme value (GEV) distribution.

7.5 Information retrieval

The field of information retrieval is focused on finding an effective computational approach to automate the storage and retrieval of documents. This information retrieval task has proven relevant in online search engines and library management systems. A study (Manning et al., 2009) summarized potential application areas such as search result clustering, scatter–gather, collection clustering language modeling, cluster-based retrieval, for the use of clustering methods the related analysis to the problem of information retrieval. Vries (2014) used two novel clustering algorithms (TopSig Ktree and EM-tree) to evaluate document cluster quality for large scale collections by first performing document clustering in information retrieval. The authors in Prabhu (2011) designed a technique for information retrieval using K-Means clustering. Elbattah and Molloy (2017) used K-Means to support decision-making regarding elderly healthcare in Ireland, with a particular focus on hip fracture care. Another study (Bellot and El-Bèze, 2000) applied hierarchical clustering methods to classify a set of documents retrieved by any information retrieval system. Liu and Croft performed cluster-based retrieval using K-Means in conjunction with language models to the task of information retrieval (Liu and Croft, 2004).

7.6 Aviation and automotive systems

Clustering methods have been applied to aviation and automotive-related issues. Specifically, clustering algorithms have been used to

detect a fault, manage emergencies, and proactive risk management. Li et al. (2015) investigated the applicability of cluster-based solutions to detect anomalies regarding flights aiding detection of associated risks from routine airline operations. They used DBSCAN to achieve a cluster analysis that successfully detected abnormal flights. Also, multiple kernel anomaly detection was able to identify operationally significant anomalies, surpassing the capability of exceedance detection. In a related study, authors applied trajectory clustering by deriving a framework to monitor aircraft behavior in given airspace (Gariel et al., 2011). This trajectory clustering method was achieved by using K-Means and DBSCAN algorithms. Wang and Pham proposed using ANOVA and Scheffe post hoc with a clustering analysis task to evaluate and improve service networks operated at airports (Wang and Pham, 2020). A similar study (Rose et al., 2020) applied the K-Means algorithm with a combination of 2-D mapping and t-Distributed Stochastic Neighbor Embedding (t-SNE) for a cluster post-processing routine which identifies driving factors in each cluster and building a hierarchical structure of cluster and sub-cluster labels. Mangortey et al. (2020) investigated the performance of Model-based Clustering, Self-Organizing Tree Algorithm (SOTA), Divisive Analysis (DIANA), Agglomerative Hierarchical, PAM, CLARA, K-Means clustering algorithms. The algorithms were used to group similar flights and identify abnormal operations and anomalies.

7.7 Bioinformatics

The application of clustering methods to bioinformatics can be grouped into two, namely, (i) the analysis of gene expression data generated from DNA microarray technologies and (ii) clustering processes that directly work on linear deoxyribonucleic acid (DNA) or protein sequences (Saxena et al., 2017). Gene expression analysis has benefited from clustering methods. It enables biologists to identify patterns within datasets relating to this domain. A huge amount of data on molecular biology is being generated from different experimentation in this domain. Such data consists of gene expression features, consisting of conclusions regarding the amount of the corresponding gene product and concurrent measurement of the gene expression level for thousands of genes under hundreds of conditions. Data generated in this form is often represented in a Data Matrix, in which rows are genes and columns are different experimental conditions, different tissues, consecutive time slots, or different patients. Other tasks that have benefited from clustering algorithms include: identification of homology; discovering natural structure inherent in gene expression data; knowing subtypes of cells; understanding gene functions and gene regulation; and mining useful information from noisy data (Oyelade et al., 2016; Guzzi et al., 2014).

The authors in Chopade and Sheetlani (2017) disclosed that in handling microarray data analysis, researchers employed the clustering method to group thousands of genes by their similarities of expression levels, thereby supporting the task of analyzing gene expression profiles. Similarly, it is widely reported that Genetic Convex Clustering Algorithms (GCCA) has been successfully applied to address the problem of clustering on an unbounded number of processors. A study in Sugavaneswaran (2017) stated that clustering algorithms had been used in gene expression. The genes are analyzed and grouped based on similarity in profiles using one of the widely used K-Means clustering algorithms. Lakhani et al. reported that to analyze biological sequences and group them into similar genes, clustering algorithms like evolutionary clustering, hierarchical clustering, K-Means, and bi-clustering have been successfully applied to the domain (Lakhani et al., 2015). Mishra et al. also strengthen the fact that the K-Means clustering algorithm has proven relevant in bioinformatics (Mishra et al., 2015). Research in Zahoor and Zafar (2020) proposed a warzone-inspired infiltration tactics-based optimization algorithm (ITO) to classify microarray gene expression. Nunes (2011), in a Master thesis, proposed time series-based clustering algorithms for separating and organizing unlabeled data into different groups whose signals are similar to each other in biomedical signals (Nunes, 2011). Two variants of the K-Means genetic algorithm selection process (KGA) were applied to the selection process in the GA algorithm. These algorithms can be applied to medical informatics and bioinformatics (Chehouri et al., 2017). In the study (Kordos and Blachnik, 2019), the authors attempt to overcome a genetic algorithm's limitation by applying a fuzzy c-means (FCM) clustering algorithm to reduce the chromosome size and solve the clusters border problem allowing the clusters to overlap.

7.8 Financial systems and economics

The sensitivity and level of security mechanisms built around the systems associated with banking and financial institutions require that the data generated from them be intelligently handled by automated systems. Cai et al. reported that banking and financial institutes are widely using clustering methods to capture the natural structure of data in a quest to improve their customer services and profit margins (Cai et al., 2010). Dardac and Boitan carried out a study to investigate the performance clustering analysis on the data from Romania institutions. They aimed to uncover the bank's risk profile by grouping such institutions into smaller, homogeneous clusters to assess which credit institutions have similar patterns according to their risk profile and profitability (Dardac and Boitan, 2019). In another study, the author presented research into main cluster-type methodologies for grouping data to use them in economic fields (Stefan, 2014). The study selected six macroeconomic indicators necessary to reveal a country's economic development and then applied hierarchical-based cluster methods on some sets of complex and heterogeneous data. In a related study, Brauksa (2013) applied K-Means for cluster analysis in comparing socio-economic development of different municipalities. Another study curated economic data and carried out clustering analysis using K-clustering and proposed techniques that use similarity measures for data files with nominal variables (Řezanková, 2014). In a different consideration of the applicability of clustering methods, authors in Novaliendry et al. (2015) performed clustering using K-Means to ease the process of handling extensive data of a systematic, detailed list of receipts, expenditures, and local spending within a year. An interesting improved K-Means algorithm based on the historical financial ratios was applied to analyze indicators related to economic attributes listed enterprises in Zhejiang province (Qian, 2006). A different study (Boyko et al., 2020) combined apriori and K-Means clustering algorithms to get a user behavior analysis template to predict a person location for the next month.

The banking sector is one of the key thriving sectors at the forefront of global digitization. However, numerous threats surfaced to hamper the digital banking initiative's advancement in digital banking development. Thus, clustering analysis can offer a convenient solution to such threats. One of the big threats expected in the banking sector is money laundering. The money laundering issue can be resolved by implementing Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering in the "Anti Money Laundering Regulatory Application System". In such a case, DBSCAN is employed for detecting and reporting suspicious banking transactions. However, the anti-money laundering regulatory application system has been effectively evaluated on large financial data where it has successfully detected and prevented money laundering on likely suspicious transactions (Yang et al., 2014).

Every banking institution must acknowledge the reality of the threats emanating from customers' ignorance of basic safe banking tips like giving out confidential information such as security passwords and pin to strangers, resulting in huge bank fraud. Banking institutions can apply clustering analysis to fish out such customers and give extra warning of being cautious about such activities and the overall notice provided to all the customers. K-Means++ and K-Means clustering approaches have recorded tremendous success in appraising cardholders' share vulnerable to bank scams such as skimming, fishing, fishing, etc.

(Alkhasov et al., 2015). Besides the threat of mitigation in the banking institution, clustering could also provide other vital applications such as location siting for optimal business extension. Every banking institution would like to expand the service covering locations to attract more customers, thereby growing the organizational profit margins. This can be implemented by installing e-corners and ATMs strategically to cover important economic locations and create branches at locations with excessive demand.

We can observe that the location sitting option plays a vital role in expanding the financial economy. The solution to such a task can be addressed by applying the generalized density-based clustering algorithm (GFDBSCAN) cluster algorithm (Kisore and Koteswaraiah, 2017). The contemporary world is mostly reliant on industrial relationships, and financial crises are crucial to any nation. Thus, long-term prediction of failure in the banking sector presents a significant confrontation. Happily, Fuzzy refinement domain adaptation offers a good answer to such a challenge (Behbood et al., 2013).

7.9 Robotics

Asmaa & Sadok presented a cluster-based solution to the Multi-Robot Task Allocation (MRTA) problem (Asmaa and Sadok, 2019). The authors proposed a clustering algorithm based on a dynamic distributed PSO technique and called it ACD^{2}PSO. The new clustering algorithm first groups the robot tasks into clusters using the dynamic distributed particle swarm optimization (D^{2}PSO) and, after that, allocates the robots to the clusters using the concept of multiple traveling salesman problems (MTSP). In another study, Guérin et al. (2018) addressed the problem of unsupervised robotic sorting (URS) using the combination of deep CNN feature extraction and standard clustering algorithms (K-Means, Minibatch K-Means (MBKM), Affinity Propagation, Mean Shift, Agglomerative Hierarchical Clustering, and DBSCAN) to obtain an industrial robotic system. Using hierarchical clustering, Arslan et al. (2016) addressed the problem of feedback motion planning and control to achieve a provably correct, computationally efficient coordinated multi-robot motion design. Boldt-Christmas and Wong (2015) applied to shift clustering algorithm that allows for the placement of robotic agents anywhere within the communicative range. Janati et al. applied the K-Means clustering algorithm to solve the problem of assigning tasks to robots for many tasks, and robots can handle their assigned tasks efficiently and efficiently (Janati et al., 2017).

7.10 Text mining

Text mining applications are essential to research, considering the ever-growing databases across different platforms. It is widely used in several sectors such as publishing and media; telecommunications, energy, and other services industries; information technology sector and Internet; banks, insurance and financial markets; political institutions, political analysts, public administration and legal documents; pharmaceutical and research companies and healthcare (G. & K., 2015). With this wide range of applicability, clustering analysis has also been researched and applied to the task of text mining.

The automation of extracting knowledge from text documents exploits the solutions provided by clustering algorithms to fast-track this task. In-text documents mining, clusters are built based on themes from text documents. This is often achieved by first transforming text documents into high-dimensional feature vectors based on frequency features. This results in a data matrix that contains rows and columns, where columns represent the count of one particular term. Furthermore, the processed documents are grouped based on the frequency of words and similar words in a subset of terms; these subsets are related to the theme of the documents. This may further reveal some important knowledge about the documents ranging from knowing the different themes a document may consist of and how to deduce the category to which the document may be assigned. Studies have shown that the modified variable string length genetic algorithm (MVGA) has successfully applied text clustering (Chopade and Sheetlani, 2017). K-Means clustering algorithms have also been successfully applied to analyzing news headlines across different portals by using document preprocessing techniques and creating clusters of similar news headlines (Lama, 2013).

Using an improved version of K-Means (Tunali et al., 2015) applied a spherical K-Means (SKM) algorithm named multi-cluster SKM clustering high dimensional document collections with high performance and efficiency. Similarly, Lydia et al. (2018) applied K-Means to the document clustering process task while calculating the centroid similarity, and cluster similarity was achieved with Euclidean Distance Similarity. Another study (Kalyanasundaram et al., 2015) applied K-Means to perform document clustering using supplementary information and the content for generating clusters with higher purity. The study in Sathya Priya and Priyadharshini (2012) proposed using semantic clustering and feature selection method to improve the clustering and feature selection mechanism with semantic relations of the text documents. Abualigah et al. solved the problem of text clustering by leveraging nature-inspired clustering-based algorithms, including Harmony Search (HS) Algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO) Algorithm, Ant Colony Optimization (ACO), Krill Herd Algorithm (KHA), Cuckoo Search (CS) Algorithm, Gray Wolf Optimizer (GWO), and Bat-inspired Algorithm (BA) (Abualigah et al., 2020).

7.11 Video surveillance

Text-based data may appear to be increasing in size on the internet and across different automated systems, but sometimes, the amount of information or knowledge we can discover from them might not be comparable with video files. A small video file can contain more information than text documents and other media files such as audio and images. Asad et al. in Asad et al. (2019) proposed a clustering algorithm that variation of the Hierarchical Agglomerative Clustering algorithm. The algorithm was designed to cluster face images of humans. Another study (Auslander et al., 2011) applied the combination K-Means clustering algorithm and the k-NN Localized p-value Estimator (KNN-LPE) to surveillance tasks for detecting threats in the domain of ground-based maritime video surveillance. Authors in Wu et al. (2015) formulate static video summarization as a clustering problem by using a new clustering algorithm (named Video Representation based High-Density Peaks Search (VRHDPS)) to analyze properties of video and gather similar frames into the cluster. In another related work on video surveillance, Damnjanovic et al. applied a spectral clustering algorithm that detects events from video through two types of summaries: static and dynamic (Damnjanovic et al., 2008).

7.12 Marketing

Considering the wide range of clustering algorithms available in the research domain, different studies have applied them to the problem of analyzing and predicting customers so that services are provided for them based on their requirements. In addition to this, automating market segmentation tasks, new product development, and product positioning have become possible through clustering methods.

Marketers have taken advantage of the automation of customer recommendation systems for processing reviews from customers. Clustering methods are now being widely applied to this task by using captured customer reviews and grouping them into reviews with similar preferences for market analysis. Such analysis provides markets with means for strategizing on aggressive marketing to beat or outperform their market competitors. Clustering algorithms have provided solutions to aid this process by grouping customers with overlapping preferences based on product type. K-Means and EM clustering algorithms were used in Hanumanth Sastry and PrasadaBabu (2013)

to analyze a steel's annual sales data to analyze sales volume and value concerning dependent attributes like products, customers and quantities sold. In a similar report, the author confirmed that clustering methods had been applied to finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records (Madhulatha, 2015). Rajagopal attempted to identify the high-profit, high-value and low-risk customers by one of the data mining techniques — customer clustering (Rajagopal, 2011); Örnek & Subaşıdeveloped a methodology using K-Means and E-M to identify the characteristics of customers (Örnek and Subaşı, 2011). A study (M., Hanji, & Hanji, 2014) attempted to achieve market segmentation of customers by using enhanced K-Means, which uses similarity measures for efficient segmentation of two-wheeler market data. In a similar study, Piggott (2015) designed a new airline-based market segment model with data clustering by investigating the performances of K-Means, expectation–maximization (EM), X-means, hierarchical and random clustering.

7.13 Object recognition and character recognition

A wide range of clustering methods and algorithms has been proposed for grouping views of 3D objects for object recognition in range data. This is made possible by first representing objects in terms of a library of range images of that object to make it possible to apply clustering tasks. Some studies, such as Saxena et al. (2017), Chellapilla et al. (2006) and Connell and Jain (1999), identified lexemes in handwritten text for writer independent handwriting recognition using clustering algorithms.

7.14 Data mining and big data mining

The increasing volume of data generated daily through different social media and other systems has necessitated efficient and optimal knowledge discovery from such huge unprocessed data. Some of these data-generating media have siloed data, while others have provided data integration and interoperation frameworks. Clustering methods have been deployed into such data warehouses to facilitate discovering knowledge and patterns that are not readily available. Moreover, big data processing tools like Hadoop are now widely used to support clustering operations. Data mining systems have also taken advantage of clustering to deploy applications capable of supportive mining information in detecting money laundry from magnificent banking data, detecting trends of behavior or responses from among people groups, and many more. The amount of data generated, collected and accumulated at industrial companies require effective automated knowledge discovery solutions. Benabdellah et al. (2018) investigated the appropriateness of K-Means, agglomerative hierarchical, DBSCAN and SOM to the problem of sparse datasets originating from industrial companies.

7.15 Dimensionality reduction

Data reduction has leverage on clustering techniques in reducing data dimensionality to automatically separate cases from fault cases. This has invariably supported fault detection procedures in production and plant systems. Thomas and Romagnoli reported that industries like chemical plants applied clustering methods to overcome the challenge of cutting-edge fault detection and diagnosis by applying the clustering algorithms to the organization of data into groups before training is possible (Thomas and Romagnoli, 2016). We argue that this challenge of data reduction before grouping is not limited to chemical plants alone but several industries needing to perform similar tasks. In Erdogmus and Kayaalp (2020), swarm-based, nature-inspired optimization algorithms, including GA, PSO, biogeography-based, and Gray wolf optimization algorithms, have been used for clustering iris datasets. Another study combined K-Means with nature-inspired algorithms to transform high-dimensional time series data onto a lower-dimensional space by selecting important points in the time series (Fuad, 2017). Nanda (2014) addressed the problem of efficiently handling large data related to environmental disaster management by proposing an enhancement to nature-inspired clustering algorithms by developing multi-objective and constrained approaches (Nanda, 2014). Sardar and Ansari (2018a,b) emphasized the need for maximizing the MapReduce programming paradigm for handling clustering challenges in real-world large dataset clustering through evolving new partition-based clustering (Sardar and Ansari, 2018a,b).

7.16 Data transfer through network

Nowadays, huge user-generated data are shared online via social media sites and forums. An effective transmission system is required to attain high-speed data transmission and prevent setbacks. However, the application of clustering analysis assists in such a situation. For instance, DBSCAN, AutoClass algorithm, and K-Means can be used to group similar traffic detection with transport layer statistics centered on the application's unique features as they transmit via a particular network (Erman et al., 2006). This facilitates data transfer by transferring a cluster of the same traffic and the network. It is preferred that sensor data be transmitted via the sensor network to the processing end be achieved using the lowest energy. In such an instance, hierarchical clustering has greatly impacted such tasks by significantly decreasing power consumption (Erman et al., 2006). This can be performed by classifying wireless sensor networks into clusters.

7.17 Urban development

Contemporary society is speedily accepting Internet of Things (IoT) technology. Modern society is rapidly adopting smart technology. It is pertinent that this technology goes beyond electronic devices by becoming parts of our everyday lives in such settings. As a result, we must be mindful of positioning our towns, building effective power supplies, and providing other basic needs. However, a data clustering algorithm can achieve such tasks effectively. For instance, an Ant clustering technique (K-Means) is a common algorithm that has recorded success in setting stations to sit firm towns and economic centers on roads (Meng and Liu, 2007). Moreover, the GFDBSCAN clustering technique can be employed to decide on the geographic position for establishing social amenities such as schools, hospitals, banks, electricity, etc. (Kisore & Koteswaraiah, 2017). For instance, K-Means clustering has tremendously contributed to pattern identification in online cable-dissemination supervision (Peng et al., 2013). Similarly, k-medoids clustering has been successfully employed to diagnose the transformer's fault (Zhou et al., 2017). The chaotic population registration procedure can be lessened by satellite imaging, but its high resolution triggers a spatial mismatch using ground reference data. However, convolutional expectation maximization can be employed to deal with such an issue (Pomente and Aleandri, 2017).

7.18 Privacy protection

Data security and privacy are the most crucial matters requiring consideration in the current digital globe. Reliable and data sharing security can be attained by separating data into clusters and presenting a critical cluster to investigators. The task can be performed by employing distributed clustering (Erkin et al., 2013) and K-Means clustering (Vaidya and Clifton, 2003). In addition to shared data security, it is imperative to secure information publicly available on the web. The existence of illegal web replication has been discovered in several cases. As a result, a hybrid of clustering algorithms has been effectively designed to jointly connect several scam-webs, thereby simplifying their identification and elimination (Drew and Moore, 2014). Another

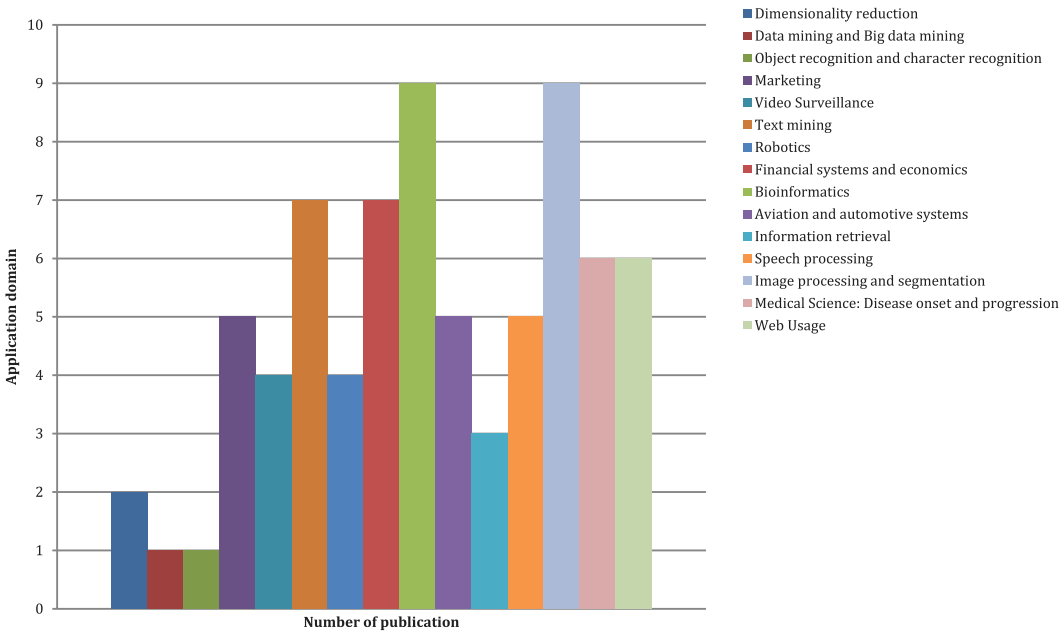


Fig. 11. A distribution of application of clustering analysis among the studies summarized in Section 6.1 through 6.14.

challenge in information security settings is identifying which family a particular malware belongs to and the signature generation of anti-virus systems. Malware that contradicts normal data is significantly regarded as outliers and, as a result, eliminated from normal data clusters. However, normal clustering techniques such as x-means and K-Means and co-clustering algorithms have been successfully employed for network anomaly detection (Ahmed et al., 2016). Behavioral malware-clustering has been made possible by exploiting hierarchical (single-linkage) clustering to secure data from attack (Biggio et al., 2014). Even though K-Means clustering has succeeded in anomaly-based intrusion detection, the modified cluster labeling algorithm and 'Opt-Grid Clustering' are more capable of attaining enhanced performance in the high-performance intrusion identification domain (Ishida et al., 2005). Clustering algorithms have made a tremendous contribution to the domain of computer security. Thus, a data clustering algorithm can deal extensively with cyber-attacks and simultaneously improve data privacy and data mining.

As reviewed in the sub-sections above, a summary of the applications of clustering methods to different domains is presented in graphical format. Our interest is to visually glance at the most applied clustering algorithm or methods and distribution of such methods' application in each domain.

The graph displayed in Fig. 11 shows the distribution of publications reviewed across the applicability of clustering algorithms. This study found that image segmentation/processing, medicine, and bioinformatics attract more research interest, applying clustering algorithms or methods to tackle data object grouping problems. This is followed by data mining and financial or economic-related application areas. The findings in this study will give prospective research work impetus to investigate the applicability of clustering algorithms to the anticipated domain and to know the extent and number the clustering algorithms are applied. This will also allow examining the application of newer or even hybrid clustering algorithms to a domain of choice depending on the trend observed, as shown in Fig. 12. Furthermore, in Fig. 13, we studied and presented how each clustering algorithm has been applied across domains. This study discovered that while some clustering algorithms enjoy an extensive range of applicability, others suffer. This could be pointers to many research opportunities for keen researchers. For instance, one could investigate a clustering algorithm's properties that make it suitable for application and enjoy adoption by many in a particular domain. This could further reveal possible improvements that could be done in that specific clustering algorithm rather than invest research efforts on a clustering algorithm that may not be profitable to that particular domain of interest.

To further study the trends indeed applying clustering methods to different domains of interest, this study plotted a graph of categorization of the use of all clustering algorithms in each domain/field. Again, this was carried out to check the peak and distribution of frequency of applicability of clustering algorithms in each group. For instance, the field of financial and economic institutions/systems attracts clustering algorithms more than other fields of human endeavors. In contrast, other clustering algorithms are least adopted for some fields. This leaves a thoughtful researcher with questions about investigating each domain's problem's characteristics versus the properties of the clustering algorithms of interest and how both maps efficiently solve the problem in the field.

.

8 Concluding remark

Clustering is a powerful data mining and analysis tool used in many fields, including machine learning, bioinformatics, robotics, pattern recognition, and image analysis. Identifying the number of clusters apriori is the most fundamental problem in cluster analysis. Specifying the correct number of clusters apriori can help obtain optimal solutions to many clustering problems. Because of this, automatic clustering algorithms are taking over traditional clustering algorithms. Automated clustering algorithms are designed to perform clustering without the prior knowledge of data sets. They also can determine the optimal number of clusters in noisy datasets. This study presents a comprehensive and up-to-date survey of traditional and state-of-the-art clustering algorithms. The paper will be beneficial for both practitioners and researchers. It outlines the strengths and weaknesses of various clustering algorithms. Moreover, it presents many open issues that interested researchers and practitioners can explore. It also offers valuable insights into clustering algorithms' practical applications to different sectors, including medical science, image processing, robotics, aviation, automotive, financial systems, and big data mining.

The findings in this study show that the field of bioinformatics, financial systems and economics, image processing, and segmentation attracts more research interest than other domains, such as data mining and object recognition. It also shows that the K-Means algorithm, Fuzzy

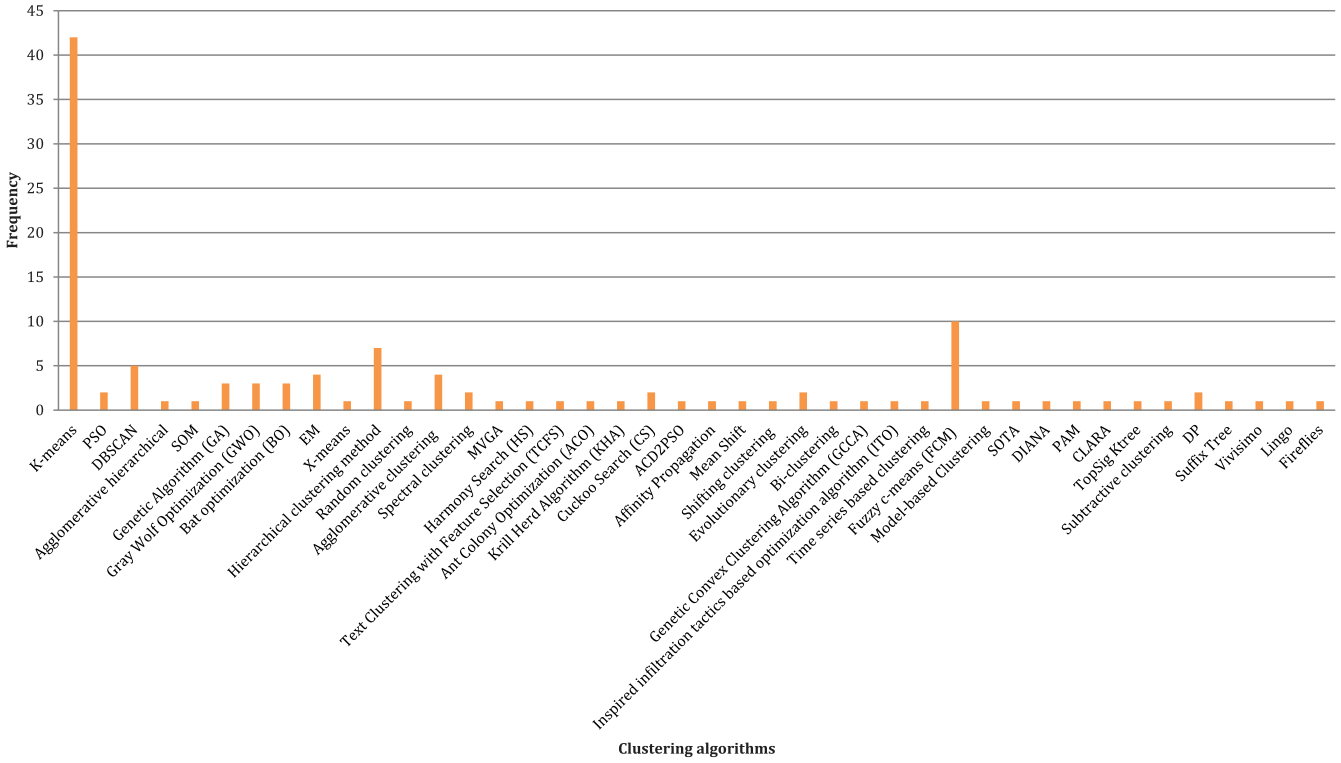


Fig. 12. An illustration of the distribution of the application of the clustering algorithms/methods across all studies considered in this section.

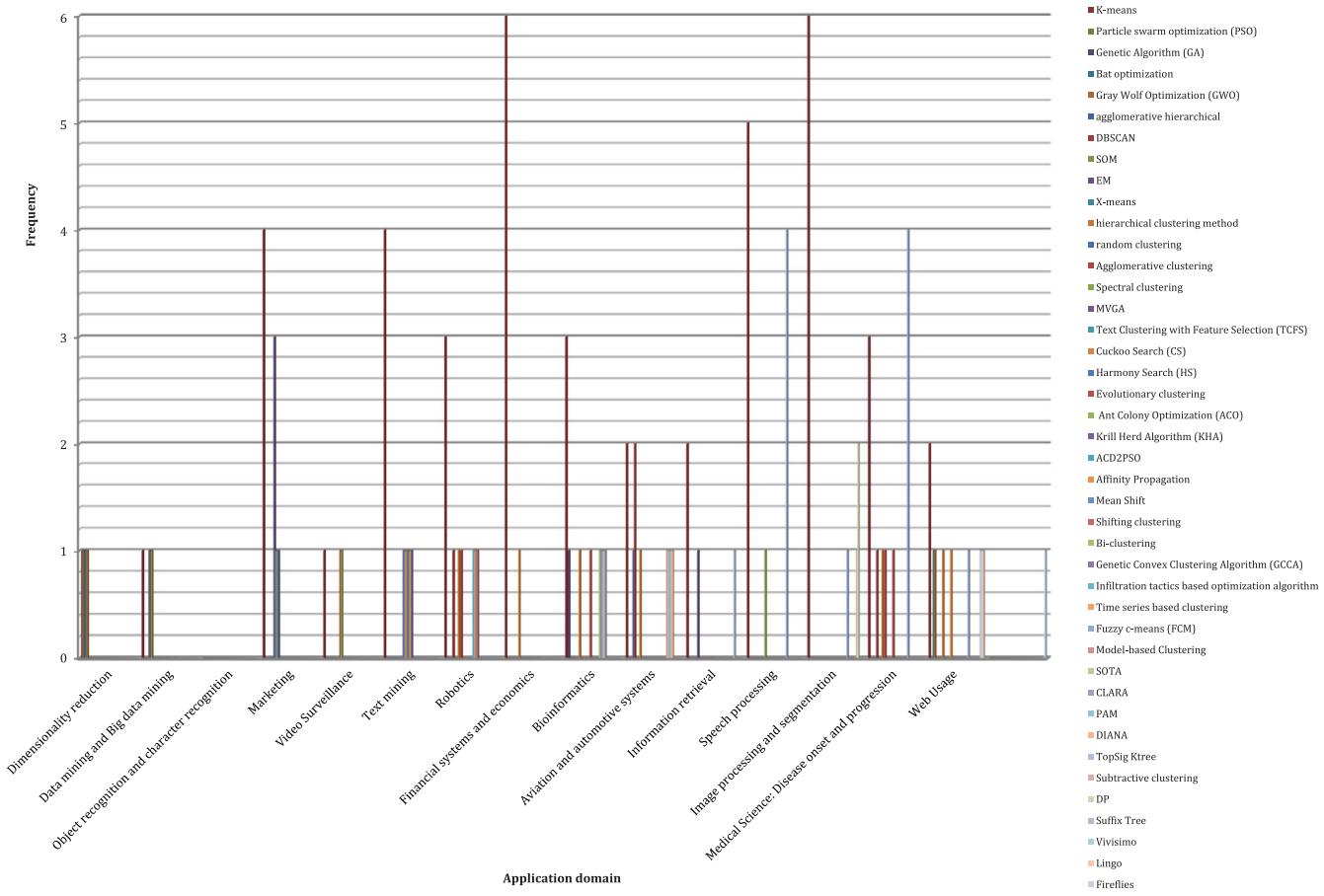


Fig. 13. Trend of the applicability of clustering algorithms/methods to different domains across all studies considered in this section.

C-Means, and Hierarchical clustering technique are among the most widely used clustering algorithms in the literature. Moreover, this study shows that many clustering algorithms (such as nature-inspired ones) have not been explored fully. These findings should give impetus for future research studies. Future research can investigate the applicability of clustering algorithms to different domains.

Furthermore, future studies can investigate the characteristics of the problems experienced in different domains versus the properties of different clustering algorithms and how both maps efficiently solve problems in other application domains. Moreover, future studies can investigate the application of newer or even hybrid clustering algorithms to a field of choice depending on various trends shown in this survey. This survey will serve as a good reference point for researchers and practitioners to design improved and efficient state-of-the-art clustering algorithms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abonyi, J., Feil, B., 2007. Cluster Analysis for Data Mining and System Identification. Birkhäuser: Springer Science & Business Media, Basel.

Abualigah, L.M.Q., 2019. Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering. Springer, Berlin, pp. 1–165.

Abualigah, L., Gandomi, A.H., Elaziz, M.A., Hussien, A.G., Khasawneh, A.M., Alshinwan, M., et al., 2020. Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis. Algorithms 13 (345), 1–32.

Abualigah, L.M., Khader, A.T., Hanandeh, E.S., 2018a. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. J. Comput. Sci. 25, 456–466.

Abualigah, L.M., Khader, A.T., Hanandeh, E.S., 2018b. Hybrid clustering analysis using improved krill herd algorithm. Appl. Intell. 48 (11), 4047–4071.

Ackermann, Marcel R., Martens, Marcus, Raupach, Christoph, Swierkot, Kamil, Lammersen, Christiane, Streamkm, Christian Sohler, 2012. A clustering algorithm for data streams. ACM J. Exp. Algorithmics 17 (4).

Adil, Fahad, Najlaa, Alshatri, Zahir, Tari, Abdullah, Alamri, Ibrahim, Khalil, Albert, Zomaya, Sebti, Foufou, Abdelaziz, Bouras, 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE Trans. Emerg. Top. Comput. 2, 267–279. http://dx.doi.org/10.1109/TETC.2014.2330519.

Agarwal, P., Alam, M.A., Biswas, R., 2011. Issues, challenges and tools of clustering algorithms. arXiv preprint arXiv:1110.2610.

Aggarwal, C.C., Hinneburg, A., Keim, D.A., 2000. On the Surprising Behavior of Distance Metrics in High Imensional Space. IBM Research report, RC 21739.

Aggarwal, C.C., Philip, S.Y., Han, J., Wang, J., 2003. A framework for clustering evolving data streams. In: Proceedings 2003 VLDB Conference. Morgan Kaufmann, pp. 81–92.

Aggarwal, C.C., Procopiuc, C., Wolf, J.L., Yu, P.S., Park, J.S., 1999. Fast algorithms for projected clustering. In: Proceedings of the ACM SIGMOD Conference, 61-72, Philadelphia, PA.

Aggarwal, C.C., Zhai, C., 2012. A survey of text clustering algorithms. In: Mining Text Data. Springer, Boston, MA, pp. 77–128.

Aghabozorgi, Saeed, Shirkhorshidi, Ali Seyed, Wah, Teh Ying, 2015. Time-series clustering – A decade review. Inf. Syst. 53, 16–38.

Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., 1998. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the ACM SIGMOD Conference, 94-105, Seattle, WA.

Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P., 2005. Automatic subspace clustering of high dimensional data. Data Min. Knowl. Discov. 11 (1), 5–33.

Ahmad, A., Dey, L., 2007. A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. 63 (2), 503–527.

Ahmed, T., Shaffer, P., Connelly, K., Crandall, D., Kapadia, A., 2016. Addressing physical safety, security, and privacy for people with visual impairments. In: Twelfth Symposium on Usable Privacy and Security (SOUPS 2016, Denver, Colorado, pp. 341–354.

Ai, L., Gao, X., Xiong, J., 2014. Application of mean-shift clustering to Blood oxygen level dependent functional MRI activation detection. BMC Med. Imaging 14 (1), 1–10.

Aitkin, M., Rubin, D., 1985. Estimation and hypothesis testing in finite mixture models. J. R. Stat. Soc. B 47, 67–75.

Akram, M.U., Khalid, S., Tariq, A., Javed, M.Y., 2013. Detection of neovascularization in retinal images using multivariate m-mediods based classifier. Comput. Med. Imaging Graph. 37 (5–6), 346–357.

Al-Jabery, K., Obafemi-Ajayi, T., Olbricht, G., Wunsch, D., 2019. Computational Learning Approaches to Data Analytics in Biomedical Applications. Academic Press.

Alguwaizani, A., Hansen, P., Mladenović, N., Ngai, E., 2011. Variable neighborhood search for harmonic means clustering. Appl. Math. Model. 35 (6), 2688–2694.

Aliniya, Z., Mirroshandel, S.A., 2019. A novel combinatorial mergesplit approach for automatic clustering using imperialist competitive algorithm. Expert Syst. Appl. 117, 243–266.

Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., Cremers, D., 2018. Clustering with deep learning: Taxonomy and new methods.

Alkhasov, S.S., Tselykh, A.N., Tselykh, A.A., 2015. Application of cluster analysis for the assessment of the share of fraud victims among bank card holders. In: Proceedings of the 8th International Conference on Security of Information and Networks. Sochi Russia, pp. 103–106, September 8 - 10. http://dx.doi.org/10.1145/2799979.2800033.

Alshamiri, A.K., Surampudi, B.R., Singh, A., 2015. A novel ELM K-means algorithm for clustering. In: Panigrahi, B., Suganthan, P., Das, S. (Eds.), Swarm, Evolutionary, and Memetic Computing. SEMCCO 2014. In: Lecture Notes in Computer Science, vol. 8947, Springer, Cham, http://dx.doi.org/10.1007/978-3-319-20294-5_19.

Amini, A., Wah, T.Y., Teh, Y.W., 2012. DENGRIS-Stream: A density-grid based clustering algorithm for evolving data streams over sliding window. In: Proc. International Conference on Data Mining and Computer Engineering. pp. 206–210.

Anter, A.M., Hassenian, A.E., Oliva, D., 2019. An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural. Expert Syst. Appl. 118, 340–354. http://dx.doi.org/10.1016/j.eswa.2018.10.009.

Antunes, C., Oliveira, A.L., 2001. Temporal data mining: an overview. In: KDD Workshop on Temporal Data Mining. pp. 1–13.

Aouf, M., Lyanage, L., Hansen, S., 2008. Review of data mining clustering techniques to analyze data with high dimensionality as applied in gene expression data (june 2008). In: 2008 International Conference on Service Systems and Service Management. IEEE, Melbourne, VIC, Australia, pp. 1–5.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., PéRez, J.M., Perona, I., 2013. An extensive comparative study of cluster validity indices. Pattern Recognit. 46 (1), 243–256.

Arslan, O., Guralnik, D.P., Koditschek, D.E., 2016. Clustering-based robot navigation and control. In: 2016 IEEE International, Conference on Robotics and Automation, Workshop on Emerging Topological Techniques in Robotics.

Asad, M.-U., Mustafa, R., Hossain, M.S., 2019. An efficient strategy for face clustering use in video surveillance system. In: Joint 2019 8th International Conference on Informatics, Electronics & Vision (ICIEV) & 3rd International Conference on Imaging, Vision & Pattern Recognition (IVPR). pp. 12–17.

Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Comput. Struct. 169, 1–12. http://dx.doi.org/10.1016/J.COMPSTRUC.2016.03.001.

Asmaa, A., Sadok, B., 2019. PSO-based dynamic distributed algorithm for automatic task clustering in a robotic swarm. In: Procedia Computer Science, 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Vol. 159. pp. 1103–1112.

Auslander, B., Gupta, K.M., Aha, D.W., 2011. A Comparative Evaluation of Anomaly Detection Algorithms for Maritime Video Surveillance. Knexus Research Corporation, Springfield, pp. 1–15.

Bach, F.R., Jordan, M.I., 2006. Learning spectral clustering, with application to speech separation. J. Mach. Learn. Res. 1963–2001.

Baker, F.B., Hubert, L.J., 1975. Measuring the power of hierarchical cluster analysis. J. Amer. Statist. Assoc. 70, 31–38.

Balavand, A., Kashan, A.H., Saghaei, A., 2018. Automatic clustering based on crow search algorithm-kmeans (CSA-Kmeans) and data envelopment analysis (DEA). In: International Journal of Computational Intelligence Systems, Vol. 11.

Ball, G.H., Hall, D.J., 1965. ISODATA, a Novel Method of Data Analysis and Pattern Classification. Stanford research inst, Menlo Park CA.

Bandaru, S., Ng, A.H., Deb, K., 2017. Data mining methods for knowledge discovery in multi-objective optimization: Part A-Survey. Expert Syst. Appl. 70, 139–159. http://dx.doi.org/10.1016/j.eswa.2016.10.015.

Bandyopadhyay, S., Saha, S., 2008. A point symmetry-based clustering technique for automatic evolution of clusters. IEEE Trans. Knowl. Data Eng. 20, 1441–1457.

Banfield, J.D., Raftery, A.E., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 80, 3–821.

Behbood, V., Lu, J., Zhang, G., 2013. Fuzzy refinement domain adaptation for long term prediction in banking ecosystem. IEEE Trans. Ind. Inf. 10 (2), 1637–1646.

Behzadi, S., Müller, N.S., Plant, C., Böhm, C., 2020. Clustering of mixed-type data considering concept hierarchies: problem specification and algorithm. Int. J. Data Sci. Anal. 10 (3), 233–248.

Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. 7 (11).

Bellot, P., El-Bèze, M., 2000. A clustering method for information retrieval.

Benabdellah, A.C., Benghabrit, A., Bouhaddou, I., 2018. A survey of clustering algorithms for an industrial context. In: Procedia Computer Science, Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018), Vol. 148. pp. 291–302.

Benabdellah, Abla Chouni, Benghabrit, Asmaa, Bouhaddou, Imane, 2019. A survey of clustering algorithms for an industrial context. Procedia Comput. Sci. (ISSN: 1877-0509) 148, 291–302.

Berkhin, P., Beche, J.D., Randall, D.J., 2001. Interactive path analysis of web site traffic. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 414–419. http://dx.doi.org/10.1145/502512.502574.

Bezdek, J.C., 2013. Pattern Recognition with Fuzzy Objective Function Algorithms. Springer Science & Business Media. Springer, Berlin.

Bezdek, J.C., Pal, N.R., 1998. Some new indexes of cluster validity. IEEE Trans. Syst. Man Cybern. B 28 (3), 301–315.

Bhattacharjee, P., Mitra, P., 2021. A survey of density based clustering algorithms. Front. Comput. Sci. 15 (1), 1–27.

Bickel, S., Scheffer, T., 2004. Multi-view clustering. In: ICDM, Vol. 4. pp. 19–26.

Biggio, B., Corona, I., Nelson, B., Rubinstein, B.I., Maiorca, D., Fumera, G., Roli, F., 2014. Security evaluation of support vector machines in adversarial environments. In: Support Vector Machines Applications. Springer, Cham, pp. 105–153.

Bindra, K., Mishra, A., 2017. A detailed study of clustering algorithms. In: 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Noida, pp. 371–376. http://dx.doi.org/10.1109/ICRITO.2017.8342454.

Boldt-Christmas, a., Wong, b., 2015. A Study of Algorithms for 2-Dimensional Self-Assembly in Swarm Robotics. Degree Project, in Computer Science, kth Royal Institute of Technology, Stockholm, Sweden.

Boley, D., 1998. Principal direction divisive partitioning. Data Min. Knowl. Discov. 2 (4), 325–344.

Bora, D.J., Gupta, A.K., 2014. A novel approach towards clustering based image segmentation. Int. J. Emerg. Sci. Eng. (IJESE) 2 (11), 6–10.

Boriah, S., Chandola, V., Kumar, V., 2008. Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp. 243–254.

Borlea, I.D., Precup, R.E., Borlea, A.B., Iercan, D., 2021. A unified form of fuzzy C-means and K-means algorithms and its partitional implementation. Knowl.-Based Syst. 214, http://dx.doi.org/10.1016/j.knosys.2020.106731.

Bouveyron, C., Hammer, B., Villmann, T., 2012. Recent developments in clustering algorithms. In: ESANN 2012 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. pp. 447–458.

Boyko, N., Komarnytska, H., Kryvenchuk, Y., Malynovskyy, Y., 2020. Clustering algorithms for economic and psychological analysis of human behavior. In: CMiGIN-2019: International Workshop on Conflict Management in Global Information Networks. pp. 1–13.

Brauksa, I., 2013. Use of cluster analysis in exploring economic indicator differences among regions: The case of latvia. J. Econ. Bus. Manage. 1 (1), 42–45.

Brimberg, J., Janićijević, S., Mladenović, N., Urošević, D., 2017. Solving the clique partitioning problem as a maximally diverse grouping problem. Optim. Lett. 11 (6), 1123–1135.

Brimberg, J., Mladenović, N., Todosijević, R., et al., 2019. Solving the capacitated clustering problem with variable neighbourhood search. Ann. Oper. Res. 272, 289–321. http://dx.doi.org/10.1007/s10479-017-2601-5.

Brito, P., Chavent, M., 2012. Divisive monothetic clustering for interval and histogram-valued data.

Cai, F., Le-Khac, N.-A., Kechadi, M.-T., 2010. Clustering approaches for financial data analysis: a survey.

Calinski, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Commun. Stat. 3, 1–27.

Campbell, J.G., Fraley, C., Murtagh, F., Raftery, A.E., 1997. Linear flaw detection in woven textiles using model based clustering. Pattern Recognit. Lett. 18, 1539–1548.

Cao, F., Ester, M., Qian, W., Zhou, A., 2006. Density-based clustering over an evolving data stream with noise. https://epubs.siam.org/page/term.

Cao, Buyang, Glover, Fred, Rego, Cesar, 2015. A tabu search algorithm for cohesive clustering problems. J. Heuristics 21, http://dx.doi.org/10.1007/s10732-015-9285-2.

Carreira-Perpiñán, M.Á., 2015. A review of mean-shift algorithms for clustering. arXiv preprint arXiv:1503.00687.

Carrizosa, E., Mladenovic, N., Todosijevic, R., 2013. Variable neighbourhood search for minimum sum-of-squares clustering on networks. European J. Oper. Res. 230, 356–363.

Chan, P.K., Mahoney, M.V., 2005. Modeling multiple time series for anomaly detection. In: Proceedings of Fifth IEEE International Conference on Data Mining. pp. 90–97.

Chang, J.W., Jin, D.S., 2002. A new cell-based clustering method for large, high-dimensional data in data mining applications. In: Proceedings of the 2002 ACM Symposium on Applied Computing. ACM Press, pp. 503–507.

Chang, D.X., Zhang, X.D., Zheng, C.W., Zhang, D.M., 2010. A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem. Pattern Recognit. 43 (4), 1346–1360.

Chaouni, A., Benabdellah, Benghabrit, A., Bouhaddou, I., 2019. A survey of clustering algorithms for an industrial context. Procedia Comput. Sci. 148, 291–302.

Chavent, M., Lechevallier, Y., Briant, O., 2007. DIVCLUS-T: A monothetic divisive hierarchical clustering method. Comput. Statist. Data Anal. 52 (2), 687–701.

Chehouri, A., Younes, R., Khoder, J., Perron, J., Ilinca, A., 2017. A selection process for genetic algorithm using clustering analysis. Algorithms 10 (123), 1–15.

Chellapilla, K., Simard, P., Abdulkader, A., 2006. Allograph based writer adaptation for handwritten character recognition. In: Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft.

Chen, Yixin, Tu, Li, 2007. Density-based clustering for real-time stream data. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007.

Chena, Z., Qi, Z., Meng, F., Cui, L., Shi, Y., 2015. Image segmentation via improving clustering algorithms with density and distance. In: Procedia Computer Science, Information Technology and Quantitative Management (ITQM 2015), Vol. 55. pp. 1015–1022.

Cheng, Y., 2002. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17 (8), 790–799, (1995) Comaniciu and Meer.

Cheng, C., Fu, A., Zhang, Y., 1999. Entropy-based subspace clustering for mining numerical data. In: Proceedings of the 5th ACM SIGKDD. San Diego, CA. pp. 84-93.

Cherng, J., Lo, M., 2001. A hypergraph based clustering algorithm for spatial data sets. In: Proc. IEEE Int. Conf. Data Mining (ICDM'01), pp. 83–90.

Chiş, M., Banerjee, S., Hassanien, A.E., 2009a. Clustering time series data: an evolutionary approach. Found. Comput. Intell. 6 (1), 193–207.

Chiş, Monica, Banerjee, Soumya, Hassanien, Aboul Ella, 2009b. Clustering time series data: An evolutionary approach. In: Abraham, A., et al. (Eds.), Foundations of Comput. Intel. Vol. 6, SCI 206. Springer-Verlag, Berlin Heidelberg, pp. 193–207, springerlink.com©.

Choi, S.S., Cha, S.H., Tappert, C.C., 2010. A survey of binary similarity and distance measures. J. Syst. Cybern. Inform. 8 (1), 43–48.

Chopade, N., Sheetlani, J., 2017. Recent trends in incremental clustering: A review. IOSR J. Comput. Eng. 19 (1), 19–24.

Chowdhury, A., Bose, S., Das, S., 2011. Automatic clustering based on invasive weed optimization algorithm. In: International Conference on Swarm, Evolutionary, and Memetic Computing. Springer, Berlin, pp. 105–112.

Chowdhury, K., Chaudhuri, D., Pal, A.K., 2020. An entropy-based initialization method of K-Means clustering on the optimal number of clusters. Neural Comput. Appl. 1–18.

Comaniciu, D., Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24 (5), 603–619.

Connell, Scott D., Jain, Anil K., 1999. Writer adaptation of online handwritten models. In: Proc. 5th Int. Conf. Document Analysis and Recognition, pp. 434–437.

Consoli, S., Korst, J., Pauws, S., Geleijnse, G., 2019. Improved variable neighbourhood search heuristic for quartet clustering. In: Sifaleras, A., Salhi, S., Brimberg, J. (Eds.), Variable Neighbourhood Search. ICVNS 2018. In: Lecture Notes in Computer Science, vol. 11328, Springer, Cham, http://dx.doi.org/10.1007/978-3-030-15843-9_1.

Corter, J.E., Gluck, M.A., 1992. Explaining basic categories: Feature predictability and information. Psychol. Bull. 111 (2), 291–303.

Cui, X., Potok, T.E., 2005. Document clustering analysis based on hybrid PSO+K-Means algorithm. J. Comput. Sci. 5, 27–33.

Cui, X., Potok, T.E., Palathingal, P., 2005. Document clustering using particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, SIS 2005. IEEE Press, Piscataway, pp. 185–191.

Cura, T., 2012. A particle swarm optimization approach to clustering. Expert Syst. Appl. 39 (1), 1582–1588. http://dx.doi.org/10.1016/j.eswa.2011.07.123.

Dafir, Z., Lamari, Y., Slaoui, S.C., 2021. A survey on parallel clustering algorithms for big data. Artif. Intell. Rev. 54 (4), 2411–2443.

Dalrymple-Alford, E.C., 1970. The measurement of clustering in free recall. Psychol. Bull. 74, 32–34.

Damnjanovic, U., Fernandez, V., Izquierdo, E., Martinez, J.M., 2008. Event detection and clustering for surveillance video summarization. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services. pp. 63–66.

Dang, X.H., Lee, V., Ng, W.K., Ciptadi, A., Ong, K.L., 2009. An EM-based algorithm for clustering data streams in sliding windows. In: International Conference on Database Systems for Advanced Applications. Springer, Berlin, Heidelberg, pp. 230–235.

Dardac, N., Boitan, I.A., 2019. A cluster analysis approach for banks' risk profile: The Romanian evidence. Eur. Res. Stud. 7 (1), 109–118.

Das, S., Abraham, A., Konar, A., 2008b. Swarm intelligence algorithms in bioinformatics. Stud. Comput. Intell. (SCI) 94, 113–147.

Das, S., Abraham, A., Sarkar, S., 2006. A hybrid rough set–particle swarm algorithm for image pixel classification. http://dx.doi.org/10.1109/HIS.2006.264909, 26.

Das, S., Chowdhury, A., Abraham, A., 2009. A bacterial evolutionary algorithm for automatic data clustering. In: 2009 IEEE Congresson Evolutionary Computation. IEEE, pp. 2403–2410.

Das, G., Lin, K.I., Mannila, H., Renganathan, G., Smyth, P., 1998. Rule discovery from time series. Knowl. Discov. Data Min. 98, 16–22.

Das, A.S., Roy, S., 2008. Swarm intelligence algorithms for data clustering. In: Soft Computing for Knowledge Discovery and Data Mining. Springer, Boston, MA, USA, pp. 279–313.

Dasgupta, S., Littman, M.L., McAllester, D., 2002. PAC generalization bounds for co-training. In: Advances in Neural Information Processing Systems, Vol. 1. pp. 375–382.

Dasgupta, A., Raftery, A.E., 1998. Detecting features in spatial point processes with clutter via model-based clustering. J. Amer. Statist. Assoc. 93, 294–302.

Datta, S., Datta, S., 2003. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics 19 (4), 459–466.

Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. (2), 224–227.

Day, W.H.E., Edelsbrunner, H., 1984. Efficient algorithms for agglomerative hierarchical clustering methods. J. Classification 1, 7–24. http://dx.doi.org/10.1007/BF01890115.

Defays, D., 1977. An efficient algorithm for a complete link method. Comput. J. 20, 364–366.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood for incomplete data via the EM algorithm. J. R. Stat. Soc. B 39, 1–38.

Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chrétien, L., 1991. The dynamics of collective sorting: robot-like ants and ant-like robots. In: Meyer, J.-A., Wilson, S. (Eds.), From Animals to Animals: Proceedings of the First International Conference on Simulation of Adaptive Behavior. MIT Press, Cambridge, pp. 356–365.

Dhanachandra, N., Manglem, K., Chanu, Y.J., 2015. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. In: Procedia Computer Science, Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), Vol. 54. pp. 764–771.

Dhiman, G., Kumar, V., 2018. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. http://dx.doi.org/10.1016/j.knosys.2018.06.001.

Dizaji, K., Herandi, A., Deng, C., Cai, W., Huang, H., 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization.

Djouzi, K., Beghdad-Bey, K.A., 2019. Review of clustering algorithms for big data. In: 2019 International Conference on Networking and Advanced Systems (ICNAS). Annaba, Algeria, pp. 1–6. http://dx.doi.org/10.1109/ICNAS.2019.8807822.

Dongkuan, Xu, 2015. A comprehensive survey of clustering algorithms.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B 26 (1), 29–41.

Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. MIT Press, Cambridge.

Doval, D., Mancoridis, S., Mitchell, B.S., 1999. Automatic clustering of software systems using a genetic algorithm. In: STEP'99. Proceedings Ninth International Workshop Software Technology and Engineering Practice. IEEE, pp. 73–81.

Drew, J., Moore, T., 2014. Automatic identification of replicated criminal websites using combined clustering. In: 2014 IEEE Security and Privacy Workshops. IEEE, pp. 116–123.

Drewes, B., 2005. Some industrial applications of text mining. In: Knowledge Mining. Springer, Berlin, Heidelberg, pp. 233–238.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification. Wiley Publications.

Duin, R.P., Fred, A.L., Loog, M., P, ekalska, E., 2012c. Mode seeking clustering by KNN and mean shift evaluated. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, pp. 51–59.

Duin, R.P.W., Fred, A.L.N., Loog, M., Pękalska, E., 2012a. Mode seeking clustering by KNN and mean shift evaluated. In: Gimel'farb, G., et al. (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. SSPR/SPR 2012. In: Lecture Notes in Computer Science, vol. 7626, Springer, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-642-34166-3_6.

Duin, Robert P.W., Fred, Ana L.N., Loog, Marco, Pękalska, Elzbieta, 2012b. Mode seeking clustering by KNN and mean shift evaluated. In: farb, G.L. Gimel', et al. (Eds.), SSPR & SPR. In: LNCS, vol. 7626, Springer-Verlag Berlin Heidelberg, pp. 51–59.

Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. 3 (3), 32–57.

D'urso, P., Massari, R., 2019. Fuzzy clustering of mixed data. Inform. Sci. 505, 513–534.

Edwards, A.W., Cavalli-Sforza, L.L., 1965. A method for cluster analysis. Biometrics 362–375.

ElazizID, M.A., Al-qaness, M.A., Zaid, E.O., Lu, S., Ibrahim, R.A., Ewees, A.A., 2020. Automatic clustering method to segment COVID-19 CT images. PLoS ONE 16 (1), 1–13.

Elbattah, M., Molloy, O., 2017. Clustering-aided approach for predicting patient outcomes with application to elderly healthcare in Ireland. In: The AAAI-17 Joint Workshop on Health Intelligence WS-17-09. pp. 533–541.

Engelbrecht, A.P., 2005. Fundamentals of Computational Swarm Intelligence. John Wiley and Sons Ltd.

Erdogmus, P., Kayaalp, F., 2020. Introductory chapter: Clustering with nature-inspired optimization algorithms. In: Erdogmus, P., Kayaalp, F. (Eds.), Introduction to Data Science and Machine Learning. IntechOpen, p. 16.

Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L., 2013. Privacy-preserving distributed clustering. EURASIP J. Inf. Secur. vol. 2013 (1), 1–15. http://dx.doi.org/10.1186/1687-417X-2013-4.

Erman, J., Arlitt, M., Mahanti, A., 2006. Traffic classification using clustering algorithms. In: Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data. pp. 281–286, Pisa Italy September 11 - 15, 2006. http://dx.doi.org/10.1145/1162678.1162679.

Ezugwu, A.E., 2020a. Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. SN Appl. Sci. 2 (2), http://dx.doi.org/10.1007/s42452-020-2073-0.

Ezugwu, A.E., 2020b. Nature-inspired metaheuristic techniques for automatic clustering: a survey and performance study. SN Appl. Sci. 2 (2), http://dx.doi.org/10.1007/s42452-020-2073-0.

Ezugwu, A.E., Shukla, A.K., Agbaje, M.B., Oyelade, O.N., José-García, A., Agushaka, J.O., 2020a. Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. Neural Comput. Appl. 1–60.

Ezugwu, A.E., Shukla, A.K., Agbaje, M.B., Oyelade, O.N., José-García, A., Agushaka, J.O., 2020b. Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. Neural Comput. Appl. 1–60.

Falkenauer, E., 1998. Genetic Algorithms and Grouping Problems. John Wiley and Sons Ltd., Chichester, England.

Feng, L., Qiu, M.-H., Wang, Y.-X., Xiang, Q.-L., Yang, Y.-F., Liu, K., 2010. A fast divisive clustering algorithm using an improved discrete particle swarm optimizer. Pattern Recognit. Lett. 31 (11), 1216–1225.

Finak, G., Bashashati, A., Brinkman, R., Gottardo, R., 2009. Merging mixture components for cell population identification in flow cytometry. Adv. Bioinform. 2009, 1–12. http://dx.doi.org/10.1155/2009/247646.

Forte, J.C., Perner, A., Horst, I.C., 2019. The use of clustering algorithms in critical care research to unravel patient heterogeneity. Intensive Care Med. 45, 1025–1028.

Fortier, J., Solomon, H., 1996. Clustering procedures. In: P.R. Krishnaiah (Ed.), Proceedings ofthe Multivariate Analysis, '66. pp. 493–506.

Fowlkes, E.B., Mallows, C.L., 2010. A method for comparing two hierarchical clusterings. J. Amer. Statist. Assoc. 78 (383), 553–569.

Fraley, Chris, Raftery, Adrian E., 1998. How many clusters? Which clustering method? Answers via model-ased cluster analysis.

Friedman, J.H., Meulman, J.J., 2002. Clustering objects on subsets of attributes. http://citeseer.nj.nec.com/friedman02clustering.html.

Friedman, H.P., Rubin, J., 1967. On some invariant criteria for grouping data. J. Amer. Statist. Assoc. 62 (320), 1159–1178.

Fu, T.K., Chung, F., Ng, C.M., 2004. Financial time series segmentation based on specialized binary tree representation.

Fuad, M.M., 2017. Applying Nature-Inspired Optimization Algorithms for Selecting Important Timestamps to Reduce Time Series Dimensionality. Coventry University, pp. 1–13.

Fukunaga, K., Hostetler, L.D., 1975. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inform. Theory 21 (1), 32–40.

Gama, J., Gaber, M. (Eds.), 2007. Learning from Data Streams. Springer.

Gan, G., Ma, C., Wu, J., 2007. Data Clustering: Theory, Algorithms, and Applications. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Gariel, M., Srivastava, A.N., Feron, E., 2011. Trajectory clustering and an application to airspace monitoring. IEEE Trans. Intell. Transp. Syst. 12 (4), 1511–1524.

Gath, I., Geva, A., 1989. Unsupervised optimal fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 11 (7), 773–781.

Ge, Y., Sealfon, S.C., 2012. Flowpeaks: A fast unsupervised clustering for flow cytometry data via K-means and density peak finding. Bioinformatics 28 (15), 2052–2058. http://dx.doi.org/10.1093/bioinformatics/bts300.

Gionis, A., Mannila, H., 2003. Finding recurrent sources in sequences. In: Proceedings of the Seventh Annual International Conference on RESEARCH in Computational Molecular Biology, 2003. pp. 123–130.

Glover, Fred, 1990. Tabu search: A tutorial. Interfaces 20, 74–94. http://dx.doi.org/10.1287/inte.20.4.74, Rafael Batres, in Computer Aided Chemical Engineering, 2012.

Glover, Fred, McMillan, C., Novick, Beth, 1985. Interactive decision software and computer graphics for architectural and space planning. Ann. Oper. Res. 5, 557–573. http://dx.doi.org/10.1007/BF02023611.

Goil, S., Nagesh, H., Choudhary, A., 1999. MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets. Technical Report CPDC-TR-9906-010, Northwestern University.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, New York.

Goldbogen, J.A., Friedlaender, A.S., Calambokidis, J., McKenna, M.F., Simon, M., Nowacek, D.P., 2013. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. BioScience 63 (2), 90–100. http://dx.doi.org/10.1525/bio.2013.63.2.5.

Gong, C., Chen, H., He, W., Zhang, Z., 2017. Improved multi-objective clustering algorithm using particle swarm optimization. Plos One.

Gowda, K.C., Krishna, G., 1978. Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern Recognit. 10 (2), 105–112. http://dx.doi.org/10.1016/0031-3203(78)90018-3.

Graves, D., Pedrycz, W., 2010. Proximity fuzzy clustering and its application to time series clustering and prediction. In: Proceedings of the 2010 10th International Conference on Intelligent Systems Design and Applications ISDA10. pp. 49–54.

Grira, N., Crucianu, M., Boujemaa, N., 2005. Unsupervised and Semi-Supervised Clustering: A Brief Survey * a Review of Machine Learning Techniques for Processing Multimedia Content, Re-Port of the MUSCLE European Network of Excellence (6th Framework Programme).

Guénoche, A., Hansen, P., Jaumard, B., 1991. Efficient algorithms for divisive hierarchical clustering with the diameter criterion. J. Classification 8 (1), 5–30.

Guérin, J., Thiery, S., Nyiri, E., Gibaru, O., 2018. Unsupervised robotic sorting: Towards autonomous decision making robots. Int. J. Artif. Intell. Appl. (IJAIA) 9 (2), 81–98.

Guha, S., Mishra, N., Motwani, R., O'Callaghan, L., 2000. Clustering data streams. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science. IEEE Computer Society.

Gulhane, A., Paikrao, P.L., Chaudhari, D.S., 2012. A review of image data clustering techniques. Int. J. Soft Comput. Eng. (IJSCE) 2 (1), 212–215.

Guzzi, P.H., Masciari, E., Mazzeo, G.M., Zaniolo, C., 2014. A discussion on the biological relevance of clustering results. In: ITBAM 2014. In: LNCS, pp. 30–44.

Halkidi, M., Batistakis, Y., Vazirgiannis, M., 2002. Clustering validity checking methods: part II. ACM Sigmod Rec. 31 (3), 19–27.
Halkidi, M., Vazirgiannis, M., 2001. Clustering validity assessment: Finding the optimal partitioning of a data set. In: Proceedings 2001 IEEE International Conference on Data Mining. IEEE, pp. 187–194.
Hall, L.O., Ozyurt, I.B., Bezdek, J.C., 1999. Clustering with a genetically optimized approach. IEEE Trans. Evol. Comput..
Hamerly, G., Elkan, C., 2004. Learning the k in K-means. In: Advances in Neural Information Processing Systems, Vol. 2003. MIT Cambridge Press, pp. 281–288.
Hamilton, J.D., 1994. Time Series Analysis. Princeton University Press, p. 159.
Hancer, E., Xue, B., Zhang, M., 2020. A survey on feature selection approaches for clustering. Artif. Intell. Rev. 53 (6), 4519–4545.
Hand, D.J., Mannila, H., Smyth, P., 2001. Principles of Data Mining (Adaptive Computation and Machine Learning). MIT Press.
Handl, J., Knowles, J., 2007. An evolutionary approach to multiobjective clustering. IEEE Trans. Evol. Comput. 11 (1), 56–76.
Handl, J., Knowles, J., Dorigo, M., 2006. Ant-based clustering and topographic mapping. Artif. Life 12 (1), 35–62.
Hansen, P., 2005. Variable neighbourhood search. Search methodology. In: Bruke, E.K., Kendall, G. (Eds.), Search Metodologies. Springer, New York, NY, USA, pp. 211–238.
Hansen, P., Brimberg, J., Urosevic, D., Mladenovic, N., 2009. Solving large p-median clustering problems by primal dual variable neighbourhood search. Data Min. Knowl. Discov. 19, 351–375.
Hansen, P., Mladenovic, N., 2001. J -Means: A new local search heuristic for minimum sum-of-squares clustering. Pattern Recognit. 34, 405–413.
Hansen, P., Mladenovic, N., 2018. Variable neighbourhood search. In: Martí, R., Pardalos, P., Resende, M. (Eds.), Handbook of Heuristics. Springer, Cham, Switzerland.
Hanumanth Sastry, S., PrasadaBabu, M., 2013. Analysis & prediction of sales data in saperp system using clustering algorithms. Int. J. Comput. Sci. Inf. Technol. (IJCSITY) 1 (4), 95–109.
Harshada, S., Deshmukh, P., Ramteke, L., 2015. Comparing the techniques of cluster analysis for big data.
Hartigan, J.A., 1975. Clustering Algorithms. John willey and sons. Inc, new york.
Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A K-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) 28 (1), 100–108.
He, W., Feng, G., Wu, Q., He, T., Wan, S., Chou, J., 2011. A new method for abrupt dynamic change detection of correlated time series. Int. J. Climatol. 32 (10), 1604–1614.
He, H., Tan, Y., 2012. A two-stage genetic algorithm for automatic clustering. Neurocomputing 81, 49–59.
Herawan, T., Deris, M.M., 2009. A framework on rough set-based partitioning attribute selection. In: Paper Presented at the International Conference on Intelligent Computing.
Herawan, T., Deris, M.M., Abawajy, J.H., 2010. A rough set approach for selecting clustering attribute. Knowl.-Based Syst. 23 (3), 220–231.
Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Michigan, USA.
Hruschka, E.R., Campello, R.J., Freitas, A.A., 2009. A survey of evolutionary algorithms for clustering. IEEE Trans. Syst. Man Cybern. C 39 (2), 133–155.
Hsu, C.C., Lin, C.W., 2018. CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Trans. Multimed. 20 (2), 421–429. http://dx.doi.org/10.1109/TMM.2017.2745702.
Huang, S., Kang, Z., Xu, Z., Liu, Q., 2021. Robust deep k-means: An effective and simple method for data clustering. Pattern Recognit. 117, http://dx.doi.org/10.1016/j.patcog.2021.107996.
Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: theory and applications. Neurocomputing 70, 489–501.
Ibrahim, Osman, H., Christofides, Nicos, 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. Int. Trans. Oper. Res. (ISSN: 0969-6016) 1 (3), 317–336. http://dx.doi.org/10.1016/0969-6016(94)90032-9.
Ishida, C., Arakawa, Y., Sasase, I., Takemori, K., 2005. Forecast techniques for predicting increase or decrease of attacks using bayesian inference. In: PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2005. IEEE, Victoria, BC, Canada, pp. 450–453. http://dx.doi.org/10.1109/PACRIM.2005.1517323.
Ito, F., Hiroyasu, T., Miki, M., Yokouchi, H., 2009. Detection of preference shift timing using time-series clustering. pp. 1585–1590.
Ivancsy, R., Kovacs, F., 2006. Clustering techniques utilized in web usage mining. In: Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, pp. 237-242.
Izakian, Z., Mesgari, M.S., Abraham, A., 2016. Automated clustering of trajectory data using a particle swarm optimization. Comput. Environ. Urban Syst. 55, 55–65.
Jadhav, A.N., Gomathi, N., 2018. WGC: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering. Alex. Eng. J. 57 (3), 1569–1584. http://dx.doi.org/10.1016/j.aej.2017.04.013.
Jain, A.K., 2010. Data clustering: 50 years beyond K-Means. Pattern Recognit. Lett. 31 (8), 651–666.
Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice-Hall, Inc, Upper Saddle River, NJ.
Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surv. 31 (3), 264–323.
Janati, F., Abdollahi, F., Ghidary, S.S., Jannatifar, M., Baltes, J., Sadeghnejad, S., 2017. Multi-robot task allocation using clustering method. In: Robot Intelligence Technology and Applications, Advances in Intelligent Systems and Computing. pp. 223–247.
Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H., 2016. Variational deep embedding: An unsupervised and generative approach to clustering. arXiv preprint arXiv:1611.05148.
José-García, A., Gómez-Flores, W., 2016. Automatic clustering using nature-inspired metaheuristics: A survey. Appl. Soft Comput. 41, 192–213.
Kailing, K., Kriegel, H.P., Pryakhin, A., Schubert, M., 2004. Clustering multi-represented objects with noise. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, pp. 394–403.
Kalyanasundaram, C., Snehal, A., Gaurav, J., Jain, S., 2015. Text clustering for information retrieval system using supplementary information. Int. J. Comput. Sci. Inf. Technol. 6 (2), 1613–1615.
Kamalzadeh, H., Ahmadi, A., Mansour, S., 2020. Clustering time-series by a novel slope-based similarity measure considering particle swarm optimization.. Appl. Soft Comput. 96, 106701.
Kanade, P.M., Hall, L.O., 2004. Fuzzy ant clustering by centroid positioning. In: Proceedings of the IEEE International Conference on Fuzzy Systems, Vol. 1. IEEE Press, Piscataway, pp. 371–376.
Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y., 2000. The analysis of a simple K-Means clustering algorithm. In: Symposium on Computational Geometry. ACM, New York, 100109.
Karypis, G., Kumar, V., 2000. Multilevel k-way hypergraph partitioning. VLSI Des. 11 (3), 285–300.
Kaufman, L., Rousseeuw, P., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York.
Kaufman, L., Rousseeuw, P.J., 2009. Finding Groups in Data: An Introduction to Cluster Analysis, Vol. 344. John Wiley & Sons.
Keogh, E., Lonardi, S., Chiu, B.Y., 2002. Finding surprising patterns in a time series database in linear time and space. In: Proceedings of the Eighth ACM SIGKDD. pp. 550–556.
Khaled, S., Al-Sultan, M., Khan, Maroof, 1995. Computational experience on four algorithms for the hard clustering problem.
Kharrousheh, Adnan, Abdullah, Salwani, Nazri, Mohd Zakree Ahmad, 2011. A modified tabu search approach for the clustering problem. J. Appl. Sci. 11, 3447–3453.
Kim, J., 2009. Dissimilarity Measures for Histogram-Valued Data and Divisive Clustering of Symbolic Objects. University of Georgia, Athens, GA, USA.
Kim, J., Billard, L., 2011. A polythetic clustering process and cluster validity indexes for histogram-valued objects. Comput. Statist. Data Anal. 55 (7), 2250–2262.
Kim, J., Billard, L., 2012. Dissimilarity measures and divisive clustering for symbolic multimodal-valued data. Comput. Statist. Data Anal. 56 (9), 2795–2808.
Kim, J., Lee, W., Song, J.J., Lee, S.-B., 2017. Optimized combinatorial clustering for stochastic processes. Cluster Comput. 20 (2), 1135–1148.
Kisore, N.R., Koteswaraiah, C.B., 2017. Improving ATM coverage area using density based clustering algorithm and voronoi diagrams. Inform. Sci. 376, 1–20. http://dx.doi.org/10.1016/j.ins.2016.09.058.
Kittler, J.V., 1976. A locally sensitive method for cluster analysis. Pattern Recognit. 8 (1), 23–33.
Kokate, U., Deshpande, A., Mahalle, P., Patil, P., 2018. Review: Data stream clustering techniques, applications, and models: Comparative analysis and discussion. Big Data Cogn. Comput. 2 (32), http://dx.doi.org/10.3390/bdcc2040032.
Koontz, W.L.G., Narendra, P.M., Fukunaga, K., 1976. A graph-theoretic approach to nonparametric cluster analysis. IEEE Trans. Comput. 25, 936–944.
Kordos, M., Blachnik, M., 2019. Improving evolutionary instance selection with clustering and ensembles. In: PP-RAI2019 Conference Proceeding. pp. 302–305.
Kosters, W., Laros, J., 2007. Metrics for mining multisets. In: International Conference on Innovative Techniques and Applications of Artificial Intelligence. Springer, London, pp. 293–303.
Kovács, F., Ivancsy, R., 2006. Cluster Validity Measurement for arbitrary Shaped clustering. In: Proceeding of the 5th. WSEAS Int.Conf. on Artificial, Knowledge Engineering and Data Bases. Madrid, Spain. pp. 372–377.
Krishna, Murty, M.N., 1999. Genetic K-means algorithm. IEEE Trans. Syst. Man Cybern. B 29 (3).
Krishnapuram, R., Keller, J., 1993. A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst. 1 (2), 98–110.
Krishnasamy, G., Kulkarni, A.J., Paramesran, R., 2014. A hybrid approach for data clustering based on modified cohort intelligence and K-Means. Expert Syst. Appl. (ISSN: 0957-4174) 41 (13), 6009–6016. http://dx.doi.org/10.1016/j.eswa.2014.03.021.
Krovi, R., 1992. Genetic algorithms for clustering: a preliminary investigation. In: System Sciences, Proceedings of the Twenty-Fifth Hawaii International Conference, Volume: Iv on. pp. 540–544.
Kulkarni, A.J., Durugkar, I.P., Kumar, M., 2013. Cohort intelligence: A selfsupervised learning behavior. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1396–1400.
Kuo, R.J., Huang, Y.D., Lin, C.C., Wu, Y.H., Zulvia, F.E., 2014. Automatic kernel clustering with bee colony optimization algorithm. Inform. Sci. 283, 107–122.

Kuo, R.J., Syu, Y.J., Chen, Z.Y., Tien, F.C., 2012. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Inform. Sci. 195 (2012), 124–140. http://dx.doi.org/10.1016/j.ins.2012.01.021.

Lago-Fernández, L.F., Corbacho, F., 2010. Normality-based validation for crisp clustering. Pattern Recognit. 43 (3), 782–795.

Lakhani, J., Chowdhary, A., Harwani, D., 2015. Clustering techniques for biological sequence analysis: A review. J. Appl. Inf. Sci. 14–32.

Lakshmi, K., Karthikeyani Visalakshi, N., Shanthi, S., 2018. Data clustering using K-means based on crow search algorithm. Sādhanā 43. http://dx.doi.org/10.1007/s12046-018-0962-3S.

Lama, P., 2013. Clustering System Based on Text Mining using the K-Means Algorithm (Bachelor's thesis). (UAS), Information Technology.

Legány, C., Juhász, S., Babos, A., 2006. Cluster validity measurement techniques. In: Proceeding of the 5th. WSEAS Int. Conf. on Artificial, Knowledge Engineering and Data Bases, February 15-17. WSEAS, Madrid, Spain, pp. 388–393.

Leng, M., Lai, X., Tan, G., Xu, X., 2009. Time series representation for anomaly detection. In: Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology. In: ICCSIT 2009, vol. 2009, pp. 628–632.

Li, L., Das, S., John Hansman, R., Palacios, R., Srivastava, A.N., 2015. Analysis of flight data using clustering techniques for detecting abnormal operations. J. Aerosp. Inf. Syst. 12 (9).

Li, X., Liang, W., Zhang, X., Qing, S., Chang, P.C., 2020. A cluster validity evaluation method for dynamically determining the near-optimal number of clusters. Soft Comput. 24 (12), 9227–9241.

Li, F., Qiao, H., Zhang, B., 2018. Discriminatively boosted image clustering with fully convolutional auto-encoders. Pattern Recognit. 83, 161–173. http://dx.doi.org/10.1016/j.patcog.2018.05.019.

Liao, T.W., 2005. Clustering of time series data-a survey. Pattern Recognit. 2005. http://dx.doi.org/10.1016/j.patcog.2005.01.025.

Lin, J., Lin, H., 2009. A density-based clustering over evolving heterogeneous data stream. In: 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, Vol. 4. IEEE, pp. 275–277.

Lin, J., Vlachos, M., Keogh, E., Gunopulos, D., 2004. Iterative incremental clustering of time series. Adv. Database Technol. 521–522.

Lin, C.X., Yu, Y., Han, J., Liu, B., 2009. Hierarchical Web-Page Clustering Via in-Page and Cross-Page Link Structures. University of Illinois at Urbana-Champaign.

Liu, X., Croft, W.B., 2004. Cluster-based retrieval using language models. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. University of Sheffield, UK, pp. 186–193. http://dx.doi.org/10.1145/1008992.1009026.

Liu, Y., Wu, X., Shen, Y., 2011. Automatic clustering using genetic algorithms. Appl. Math. Comput. 218 (4), 1267–1279.

Lo, K., Brinkman, R.R., Gottardo, R., 2008. Automated gating of flow cytometry data via robust model-based clustering. Cytometry A 73 (4), 321–332. http://dx.doi.org/10.1002/cyto.a.20531.

Lu, Y., Cao, B., Rego, C., Glover, F., 2018. A tabu search based clustering algorithm and its parallel implementation on spark. Appl. Soft Comput. 63, 97–109.

Lu, Yi, Lu, Shiyong, Fotouhi, Farshad, 2004a. FGKA: A fast genetic K-means clustering algorithm. In: SAC'04 Nicosia, Cyprus. ACM, 1-58113-812-1/03/04.

Lu, Yi, Lu, Shiyong, Fotouhi, Farshad, Deng, Youping, Susan, Brown J., 2004b. An incremental genetic K-means algorithm and its application in gene expression data analysis. BMC Bioinformatics.

Lydia, E.L., Govindaswamy, P., Lakshmanaprabu, S.K., Ramya, D., 2018. Document clustering based on text mining K-means algorithm using euclidean distance similarity. J. Adv. Res. Dyn. Control Syst. 10 (02-Special Issue), E.

Macnaughton-Smith, P., Williams, W., Dale, M., Mockett, L., 1964. Dissimilarity analysis: a new technique of hierarchical sub-division. Nature 202 (4936), 1034–1035.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, No. 14. pp. 281–297.

Madhulatha, T.S., 2015. An overview on clustering methods. IOSR J. Eng. 2 (4), 719–725.

Magoeva, K., Krzhizhanovskaya, V.V., Kovalchuk, S.V., 2018. Application of clustering methods for detecting critical acute coronary syndrome patients. In: Procedia Computer Science, 7th International Young Scientist Conference on Computational Science, Vol. 138. pp. 370–379.

Mangortey, E., Monteiro, D., Ackley, J., Gao, Z., Puranik, T.G., Kirby, M., et al., 2020. Application of Machine Learning Techniques To Parameter Selection for Flight Risk Identification. Georgia Inst. Technol., Atlanta, pp. 1–39.

Manning, C.D., Raghavan, P., Schütze, H., 2009. Flat clustering. In: Clustering in Information Retrieval. Cambridge University Press, Cambridge, pp. 349–375.

Mansalis, S., Ntoutsi, E., Pelekis, N., Theodoridis, Y., 2018. An evaluation of data stream clustering algorithms. Statist. Anal. Data Min. ASA Data Sci. J. 11. http://dx.doi.org/10.1002/sam.11380.

Mao, J., Jain, A.K., 1996. A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Trans. Neural Netw. 7 (1), 16–29.

Marriot, F.H., 1975. Practical problems in a method of cluster analysis. Biometrics 27, 456–460.

Marriott, F.H.C., 1974. The Interpretation of Multiple Observations. Academic Press, London.

Martins, P., 2020. Goal clustering: VNS based heuristics. Available online: https://arxiv.org/abs/1705.07666v4 (accessed on 24 October 2020).

Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm-based clustering technique. Pattern Recognit. 33.

Mazlack, L.J., He, A., Zhu, Y., 2000. A rough set approach in choosing partitioning attributes. In: Paper Presented At the Proceedings of the ISCA 13th International Conference (CAINE-2000).

McClain, J.O., Rao, V.R., 1975. Clustisz: A program to test for the quality of clustering of a set of objects. J. Mar. Res. 456–460.

Mclachlan, G., Basford, K., 1988. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York, NY.

Mclachlan, G., Krishnan, T., 1997. The EM Algorithm and Extensions. John Wiley & Sons, New York, NY.

Meng, Y., Liu, X., 2007. Application of K-means algorithm based on ant clustering algorithm in macroscopic planning of highway transportation hub. In: 2007 First IEEE International Symposium on Information Technologies and Applications in Education. IEEE, Kunming, China, pp. 483–488. http://dx.doi.org/10.1109/ISITAE.2007.4409331.

Van der Merwe, D., Engelbrecht, A., 2003. Data clustering using particle swarm optimization. In: Proceedings of the 2003 Congress on Evolutionary Computation. IEEE Press, Piscataway, pp. 215–220.

Milligan, G.W., 1981. A Monte Carlo study of thirty internal criterion measures for cluster analysis. Psychometrika 46 (2), 187–199.

Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., Long, J., 2018. A survey of clustering with deep learning: From the perspective of network architecture. IEEE Access 6, 39501–39514. http://dx.doi.org/10.1109/ACCESS.2018.2855437.

Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Adv. Eng. Softw. 95, 51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

Mishra, J., Agarwal, V., Sharma, M., Srivastava, J., 2015. Clustering algorithms: Brief review in bioinformatics. Int. J. Sci. Res. (IJSR) 1012–1019.

Mitsa, T., 2009. Temporal Data Mining, Vol. 33. Chapman & Hall/CRC Taylor and Francis Group, Boca Raton, FL.

Mladenovic, N., Hansen, P., 1997. Variable neighbourhood search. Comput. Oper. Res. 24, 1097–1100.

Morchen, F., Ultsch, A., Mörchen, F., Hoos, O., 2005. Extracting interpretable muscle activation patterns with time series knowledge mining. J. Knowl. Based 9 (3), 197–208.

Mukerjee, S., Feigelson, E.D., Babu, G.J., Murtagh, F., Fraley, C., Raftery, A., 1998. Three types of gamma ray bursts. Astrophys. J. 508, 314–327.

Murphy, R.F., 1985. Automated identification of subpopulations in flow cytometric list mode data using cluster. In: Alan R. Liss, Inc. Cytometry, Vol. 6.

Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. Comput. J. 26, 354–359. http://dx.doi.org/10.1093/comjnl/26.4.354.

Murtagh, F., 1985. A survey of algorithms for contiguity-constrained clustering and related problems. Comput. J. 28, 82–88. http://dx.doi.org/10.1093/comjnl/28.1.82.

Myhre, J.N., Mikalsen, K.Ø., Løkse, S., Jenssen, R., 2018. Robust clustering using a kNN mode seeking ensemble. Pattern Recognit. 76, 491–505.

Nagpal, Arpita, 2013. Review based on data clustering algorithms.

Naim, I., Datta, S., Rebhahn, J., Cavenaugh, J.S., Mosmann, T.R., Sharma, G., 2014. SWIFT—scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, part 1: Algorithm design. Cytometry A 85 (5), 408–421.

Nameirakpam, D., Jina, C.Y., 2015. Image segmentation method using K-means clustering algorithm for color image. Adv. Res. Electr. Electron. Eng. 2 (11), 68–72.

Nanda, S.J., 2014. Nature inspired clustering algorithms for analysis of natural databases. Hydrol. Meteorol. 5, 4.

Nanda, S.J., Panda, G., 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol. Comput. 16, 1–18.

Nasiri, J., Farzin, Khiyabani, M., Khiyabani, F.M., 2018. A whale optimization algorithm (WOA) approach for clustering under a creative commons attribution (CC-BY) 4.0 license a whale optimization algorithm (WOA) approach for clustering. http://dx.doi.org/10.1080/25742558.2018.1483565.

Neel, J., 2005. Cluster Analysis Methods for Speech Recognition (Master Thesis in Speech Technology). Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm.

Newcomer, S.R., Steiner, J.F., Bayliss, E.A., 2011. Identifying subgroups of complex patients with cluster analysis. Am. J. Manage. Care 17 (8), 324–332.

Nirkhi, S., Hande, K., 2008. A survey on clustering algorithms for web applications. In: SWWS. pp. 124–129.

Novaliendry, D., Hendriyani, Y., Yang, C.-H., Hamimi, H., The optimized K-means clustering algorithms to analyzed the budget revenue expenditure in padang. in: Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2015), Palembang, Indonesia, pp. 61-66.

Ntoutsi, Irene, Zimek, Arthur, Palpanas, Themis, Kroger, Peer, Kriegel, Hans-Peter, 2012. Density-based projected clustering over high dimensional data streams. In: Proc. of the 12th SIAM International Conference on Data Mining.

Nunes, N.F., 2011. Algorithms for Time Series Clustering Applied to Biomedical Signals. New University of Lisbon, Faculty of Sciences and Technology, Physics Department, Lisbon.

Olson, C., 1995. Parallel algorithms for hierarchical clustering. Parallel Comput. 21, 1313–1325.
Omran, M., Salman, A., Engelbrecht, A., 2002. Image classification using particle swarm optimization. In: Wang, L., Tan, K.C., Furuhashi, T., Kim, J.-H., Yao, X. (Eds.), Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'02. IEEE Press, Piscataway, pp. 370–374.
Orlov, V., Kazakovtsev, L., Rozhnov, I., Popov, N., Fedosov, V., 2018. Variable neighbourhood search algorithm for K-means clustering. IOP Conf. Series: Mater. Sci. Eng. 450, 022035.
Ormerod, P., Mounfield, C., 2000. Localised structures in the temporal evolution of asset prices. In: New Approaches to Financial Economics. Santa Fe Conference.
Örnek, Ö., Subaşı, A., 2011. Clustering marketing datasets with data mining techniques. Journal 408–412.
Ouadfel, S., Batouche, M., Taleb-Ahmed, A., 2010. A modified particle swarm opti-mization algorithm for automatic image clustering. In: Proceedings of the 1stInternational Symposium on Modeling and Implementing Complex Systems. Constantine, Algeria.
Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghiren, E., Ameh, F., et al., 2016. Clustering algorithms: Their application to gene expression data. Bioinform. Biol. Insights 10, 237–253.
Pakhira, M.K., Bandyopadhyay, S., Maulik, U., 2004. Validity index for crisp and fuzzy clusters.. Pattern Recognit. 37 (3), 487–501.
Pandit, S., Gupta, S., 2011. A comparative study on distance measuring approaches for clustering.. Int. J. Res. Comput. Sci. 2 (1), 29–31.
Parida, P., 2018. Fuzzy clustering based transition region extraction for image segmentation. Future Comput. Inform. J. 32, 1–333.
Parmar, D., Wu, T., Blackhurst, J., 2007. Mmr: an algorithm for clustering categorical data using rough set theory. Data Knowl. Eng. 63 (3), 879–893.
Parsons, Lance, Haque, Ehtesham, Liu, Huan, 2004. Subspace clustering for high dimensional data: A review. SIGKDD Explor. 6, 90–105. http://dx.doi.org/10.1145/1007730.1007731.
Patel, P.M., Shah, B.N., Shah, V., 2013. Image segmentation using K-mean clustering for finding tumor in medical application. Int. J. Comput. Trends Technol. 4 (5), 1239–1242.
Paterlini, S., Krink, T., 2006. Differential evolution and particle swarm optimisation in partitional clustering. Comput. Stat. Data Anal. 50 (5), 1220–1247.
Patidar, A.K., Agrawal, J., Mishra, N., 2012. Analysis of different similarity measure functions and their impacts on shared nearest neighbor clustering approach. Int. J. Comput. Appl. 40 (16), 1–5.
Patil, C., Baidari, I., 2019. Estimating the optimal number of clusters k in a dataset using data depth. Data Sci. Eng. 4 (2), 132–140.
Pavlidis, N., Plagianakos, V.P., Tasoulis, D.K., Vrahatis, M.N., 2006. Financial forecasting through unsupervised clustering and neural networks. Oper. Res. 6 (2), 103–127.
Pelleg, D., 2000. Extending K-Means with efficient estimation of the number of clusters in ICML. In: Proceedings of the 17th international conference on machine learning, pp. 277–281.
Peng, X., Zhou, C., Hepburn, D.M., Judd, M.D., Siew, W.H., 2013. Application of K-means method to pattern recognition in on-line cable partial discharge monitoring. IEEE Trans. Dielectr. Electr. Insul. 20 (3), 754–761.
Piggott, J., 2015. Identification of Business Travelers Through Clustering Algorithms (Master thesis Business and Information Technology). Faculty of Management and Governance, Faculty of Electrical Engineering, Mathematics and Computer Science.
Pitchai, R., Supraja, P., Sulthana, R., Veeramakali, T., 2021. Brain tumor segmentation and prediction using fuzzy neighborhood learning approach for 3D MRI images.
Plant, C., Böhm, C., 2009. Novel Trends in Clustering. Technische Universität München, Munich Germany, pp. 1–28.
Pomente, A., Aleandri, D., 2017. Convolutional expectation maximization for population estimation. In: CLEF (Working Notes). http://ceur-ws.org/Vol-1866/paper_138.pdf. (Accessed July 7, 2021).
Popivanov, I., Miller, R.J., 2002. Similarity search over time series data using wavelets. In: Proceedings of the 18th Int'l Conference on Data Engineering, San Jose, CA, Feburary 26-March 1, pp. 212–221.
Prabhu, P., 2011. Document clustering for information retrieval – a general perspective. Indian Streams Res. J. 1–4.
Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.M., 2002. A monte carlo algorithm for fast projective clustering. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. ACM Press, pp. 418–427.
Punit, R., 2018. Big Data Cluster Analysis and its Applications (Doctor of Philosophy). The University Of Melbourne.
Qian, Y., 2006. K-means algorithm and its application for clustering companies listed in zhejiang province. Data Min. VII: Data Text Web Min. Bus. Appl. 35–44.
Qin, H., Ma, X., Herawan, T., Zain, J.M., 2014. Mgr: An information theory based hierarchical divisive clustering algorithm for categorical data. Knowl.-Based Syst. 67, 401–411.
Qu, J., Shao, Z., Liu, X., 2010. Mixed PSO clustering algorithm using point symmetry distance. J. Comput. Inf. Syst. 6 (6), 2027–2035.
Raftery, A., 1986. A note on Bayes factors for log-linear contingency table models with vague prior information. J. R. Statist. Soc. 48 (2), 249–250.
Ragaventhiran, J., Kavithadevi, M.K., 2020. Map-optimize-reduce: CAN tree assisted FP-growth algorithm for clusters based FP mining on hadoop. Future Gener. Comput. Syst. 103, 111–122. http://dx.doi.org/10.1016/j.future.2019.09.041.
Rahnema, N., Gharehchopogh, F.S., 2020. An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. Multimedia Tools Appl. 79 (43–44), 32169–32194. http://dx.doi.org/10.1007/s11042-020-09639-2.
Rajagopal, S., 2011. Customer data clustering using data mining technique. Int. J. Database Manage. Syst. (IJDMS) 3 (4).
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. J. Amer. Statist. Assoc. 66 (336), 846–850.
Rani, S., Sikka, G., 2012. Recent techniques of clustering of time series data: a survey. Int. J. Comput. Appl. 52 (15).
Räsänen, O., 2007. Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture. Helsinki University of Technology, Department of Electrical and Communications Engineering, Laboratory of Acoustics and Audio Signal Processing.
Rasmussen, E.M., 1992. Clustering Algorithms. Information Retrieval: Data Structures & Algorithms, vol. 419. p. 442.
Rastgarpour, M., Shanbehzadeh, J., Soltanian-Zadeh, H., 2014. A hybrid method based on fuzzy clustering and local region-based level set for segmentation of inhomogeneous medical images. J. Med. Syst. 38 (8), 1–15.
Ratkowsky, D.A., Lance, G.N., 1978. A criterion for determining the number of groups in a classification. Aust. Comput. J. 10, 115–117.
Ray, S., Turi, R.H., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques. Narosa Pub. House, Calcutta, pp. 137–143.
Reddy, M., Babu, M.R., 2019. Implementing self adaptiveness in whale optimization for cluster head section in Internet of Things. Cluster Comput. 22 (1), 1361–1372.
Ren, J., Cai, B., Hu, C., 2011. Clustering over data streams based on grid density and index tree. J. Converg. Inf. Technol. 6 (1), 83–93.
Ren, J., Ma, R., 2009. Density-based data streams clustering over sliding windows. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 5. IEEE, pp. 248–252.
Rendon, E., Garcia, R., Abundez, I., Gutierrez, C., Gasca, E., Del Razo, F., Gonzalez, A., 2008. Niva:A robust cluster validity. In: WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering. World Scientific and Engineering Academy and Society, Crete, Greece, pp. 209–213.
Řezanková, H., 2014. Cluster analysis of economic data. STATISTIKA 94 (1), 73–86.
Rijsbergen, V., 1979. Information Retrieval. Butterworths, London.
Rohlf, F.J., 1974. Methods of comparing classifications. Annu. Rev. Ecol. Syst. 5 (1), 101–113.
Rokach, L., 2005. Clustering Methods, Data Mining and Knowledge Discovery Handbook. Springer, pp. 331–352.
Ros, F., Guillaume, S., 2019. Information sciences munec: A mutual neighbor-based clustering algorithm. In: Information Sciences, Vol. 00. https://www.elsevier.com/open-access/userlicense/1.0/.
Rose, R.L., Puranik, T.G., Mavris, D.N., 2020. Natural language processing based method for clustering and analysis of aviation safety narratives. Aerospace 7 (143), 1–22.
Rousseeuw, P., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.
Rovetta, S., Mnasri, Z., Masulli, F., Cabri, A., 2019. Emotion recognition from speech signal using fuzzy clustering. In: Atlantis Studies in Uncertainty Modelling, 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019). pp. 120–127.
Rozhnov, I.P., Orlov, V.I., Kazakovtsev, L.A., 2019. Vns-based algorithms for the centroid-based clustering problem. Facta Univ. Ser. Math. Inform. 34, 957–972.
Ruiz, Carlos, Menasalvas, Ernestina, Spiliopoulou, Myra, C-denstream 2009. Using domain knowledge on a data stream. In: Proc. of the International Conference on Information Engineering and Computer Science, ICIECS.
Runkler, T.A., 2005. Ant colony optimization of clustering models. Int. J. Intell. Syst. 20 (12), 1233–1251.
Saatchi, S., Hung, C.C., 2005. Hybridization of the ant colony optimization with the K-means algorithm for clustering. In: Image Analysis. In: Lecture notes in computer science, vol. 3540, Springer, Berlin.
Saemi, B., Hosseinabadi, A.A., Kardgar, M., Balas, V.E., Ebadi, H., 2018. Nature inspired partitioning clustering algorithms: A review and analysis. Soft Comput. Appl. Adv. Intell. Syst. Comput. 643, 97–116.
Safari, Z., Mursi, K.T., Zhuang, Y., 2020. Fast automatic determination of cluster numbers for high dimensional big data. In: Proceedings of the 2020 the 4th International Conference on Compute and Data Analysis, pp. 50–57.
Saha, S., Alok, A.K., Ekbal, A., 2016. Brain image segmentation using semi-supervised clustering. Expert Syst. Appl. 52, 50–63.
Saitta, R.B., Smith, I., 2007. Abounded index for cluster validity. In: Perner, P. (Ed.), In: Machine Learning and Data Mining in Pattern Recognition Lecture Notes in Computer Science, vol. 4571, Springer, Berlin, Heidelberg, pp. 174–187.
Sanse, K., Sharma, M., 2015a. Clustering methods for big data analysis. Int. J. Adv. Res. Comput. Eng. Technol. 4 (3), 642–648.
Sanse, Keshav, Sharma, Meena, 2015b. Clustering methods for Big data analysis.
Sardar, T.H., Ansari, Z., 2018a. An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm. Future Comput. Inform. J. 3, 200–209.

Sardar, T.H., Ansari, Z., 2018b. Partition based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions. Future Comput. Inform. J. 3, 143–151.

Sasaki, Hiroaki, Kanamori, Takafumi, Hyvarinen, Aapo, Niu, Gang, Sugiyama, Masashi, 2018. Mode-seeking clustering and density ridge estimation via direct estimation of density-derivative-ratios. J. Mach. Learn. Res. 18 (2018), 1–47.

Sathya Priya, P., Priyadharshini, S., 2012. Clustering technique in data mining for text documents. Int. J. Comput. Sci. Inf. Technol. 294, 3–2947.

Savaresi, S.M., Boley, D.L., Bittanti, S., Gazzaniga, G., 2002. Cluster selection in divisive clustering algorithms. In: Paper Presented at the Proceedings of the 2002 SIAM International Conference on Data Mining.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., et al., 2017. A review of clustering techniques and developments. Neurocomputing http://dx.doi.org/10.1016/j.neucom.2017.06.053.

Scott, A.J., Symons, M.J., 1971. Clustering methods based on likelihood ratio criteria. Biometrics 387–397.

Senthilnath, J., Omkar, S.N., Mani, V., 2011. Clustering using firefly algorithm: performance study. Swarm Evol. Comput. 1 (3), 164–171.

Sfetsos, A., Siriopoulos, C., 2004. Time series forecasting with a hybrid clustering scheme and pattern recognition. IEEE Trans. Syst. Man Cybern. 34 (3), 399–405.

Shaffer, E., Dubes, R.C., Jain, A.K., 1979. Single-link characteristics of a mode-seeking clustering algorithm. Pattern Recognit. 11 (1), 65–70.

Sharan, R., Shamir, R., 2000. CLICK: a clustering algorithm with applications to gene expression analysis. In: Proc international conference intelligent systems molecular biology, pp. 307–316.

Sharma, S., 1996. Applied Multivariate Techniques. John Willwy & Sons.

Sheikh, R.H., Raghuwanshi, M.M., Jaiswal, A.N., 2008. Genetic algorithm based clustering: a survey. In: First International Conference on Emerging Trends in Engineering and Technology, Vol. 2. IEEE, Nagpur, India, pp. 314–319.

Shekar, B., Murty, M.N., Krishna, G., 1987. A knowledge-based clustering scheme. Pattern Recognit. Lett. 5 (4), 253–259.

Shi, Z., Wu, D., Guo, C., Zhao, C., Cui, Y., Wang, F.Y., 2021. FCM-RDpA: TSK fuzzy regression model construction using fuzzy C-means clustering, regularization, Droprule, and Powerball Adabelief. Inform. Sci. 574, 490–504. http://dx.doi.org/10.1016/j.ins.2021.05.084.

Shirkhorshidi, A.S., Shirkhorshidi, A.S., Wah, T.Y., Herawan, T., 2014. Big data clustering: A review. In: ICCSA 2014, Part V. In: LNCS 8583, Springer International Publishing Switzerland, pp. 707–720.

Sibson, R., 1973. SLINK: An optimally efficient algorithm for the single-link cluster method. Comput. J. 16 (1), 30–34. http://dx.doi.org/10.1093/comjnl/16.1.30.

Silva, J.A., Faria, E.R., Barros, R.C., Hruschka, E.R., Carvalho, A.C.D., Gama, J., 2013. Data stream clustering: A survey. ACM Comput. Surv. 46 (1), 1–31.

Sinaga, K.P., Yang, M.S., 2020. Unsupervised K-means clustering algorithm. IEEE Access 8, 80716–80727.

Singh, S., Srivastava, S., 2020. Review of clustering techniques in control system. Procedia Comput. Sci. 173, 272–280. http://dx.doi.org/10.1016/j.procs.2020.06.032.

Singh, A., Yadav, A., Rana, A., 2013. K-means with three different distance metrics. Int. J. Comput. Appl. 67 (10).

Sneath, P.H., Sokal, R.R., 1973. Numerical taxonomy. In: The Principles and Practice of Numerical Classification.

Sonkamble, B.A., Doye, D.D., 2012. Speech recognition using vector quantization through modified K-MeansLBG Algorithm. Comput. Eng. Intell. Syst. 3 (7), 137–145.

Soppari, K., Chandra, N.S., 2020. Development of improved whale optimization-based FCM clustering for image watermarking. Comp. Sci. Rev. 37, 100287. http://dx.doi.org/10.1016/J.COSREV.2020.100287.

Sruthi, K., Reddy, M.B., 2013. Document clustering on various similarity measures. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 3 (8), 1269–1273.

Stefan, R.-M., 2014. Cluster type methodology for grouping data. Proc. Econ. Finance Emerg. Markets Queries Finance Bus. 15, 357–362.

Steinback, M., Tan, P.N., Kumar, V., Klooster, S., Potter, C., 2002. Temporal data mining for the discovery and analysis of ocean climate indices. In: The 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Alberta, Canada.

Strehl, A., Ghosh, J., 2000. Clustering guidance and quality evaluation using relationship-based visualization. In: Intelligent Engineering Systems through Artificial Neural Networks. St. Louis, Missouri, USA, pp. 483–488.

Su, Z.G., Wang, P.H., Shen, J., Li, Y.G., Zhang, Y.F., Hu, E.J., 2012. Automatic fuzzy partitioning approach using variable string length artificial bee colony (VABC) algorithm. Appl. Soft Comput. 12 (11), 3421–3441.

Suganya, R., Pavithra, M., Nandhini, P., 2018. Algorithms and challenges in big data clustering. Int. J. Eng. Tech. 4 (4), 40–47.

Sugavaneswaran, L., 2017. Mathematical modeling of gene networks. Encycl. Biomed. Eng. 1–23. http://dx.doi.org/10.1016/B978-0-12-801238-3.64118-1.

Sung, C.S., Jin, H.W., 2000. A tabu-search-based heuristic for clustering. Pattern Recognit. 33, 849–858. http://dx.doi.org/10.1016/S0031-3203(99)00090-4.

Suresh, K., Kundu, D., Ghosh, S., Das, S., Abraham, A., 2009. Data clustering using multi-objective differential evolution algorithms. Fund. Inform. 97 (4), 381–403.

Tan, P.N., 2018. Introduction to Data Mining. Pearson Education India.

Tang Rui, S.F., Yang, X.-S., Deb, S., 2012. Nature-inspired clustering algorithms for web intelligence data. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. pp. 147–153.

Theodoridis, S., Koutroubas, K., 1999. Pattern Recognition. Academic Press.

Thilakaratne, M., Falkner, K., Atapattu, T., 2019. A systematic review on literature-based discovery workflow [PeerJ]. PeerJ Computer Science, 5, e235.trees. Pattern Recognit. 43, 752–766.

Thomas, M.C., Romagnoli, J., 2016. Extracting knowledge from historical databases for process monitoring using feature extraction and data clustering. In: Proceedings of the 26th European Symposium on Computer Aided Process Engineering – ESCAPE 26, 859-864.

Tran, L., Fan, L., Shahabi, C., 2021. Mixed-type data with correlation-preserving embedding. In: International Conference on Database Systems for Advanced Applications. Springer, Cham, pp. 342–358.

Tsay, R.S., 2005. Analysis of Financial Time Series. John Wiley & SONS.

Tunali, V., Bilgin, T., Camurcu, A., 2015. An improved clustering algorithm for text mining: Multi-cluster spherical K-means. Int. Arab J. Inform. Technol. 1, 2–19.

Ultsch, A., Mörchen, F., 2005. ESOM-Maps: Tools for Clustering, Visualization, and Classification with Emergent SOM.

Vaidya, J., Clifton, C., 2003. Privacy-preserving k-means clustering over vertically partitioned data. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 206–215.

Vani, H., Anusuya, M.A., 2019. Fuzzy clustering algorithms - comparative studies for noisy speech signals. Ictact J. Soft Comput. 9 (3), 1920–1926.

Venkataramana, B., Padmasree, L., Srinivasa Rao, M., Ganesan, G., Rama Krishna, K., 2017. Implementation of clustering algorithms for real datasets in medical diagnostics using MATLAB. J. Soft Comput. Appl. 2017 (1), 53–66.

Verbeek, J., 2004. Mixture Models for Clustering and Dimension Reduction (Doctoral dissertation). Universiteit van Amsterdam.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A., 2008 Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pp. 1096–1103.

Vlachos, M., Gunopulos, D., Das, G., 2004. Indexing time-series under conditions of noise. In: Last, M., Kandel, A., Bunke, H. (Eds.), Data Mining in Time Series Databases. World Scientific, Singapore, p. 67.

Voorhees, Ellen M., 1986. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Inf. Process. Manage. (ISSN: 0306-4573) 22 (6), 465–476.

Vries, C.M., 2014. Document clustering algorithms, representations and evaluation for information retrieval. In: Computational Intelligence and Signal Processing. Queensland University of Technology, Brisbane, Australia.

Žalik, K.R., Žalik, B., 2011. Validity index for clusters of different sizes and densities. Pattern Recognit. Lett. 32 (2), 221–234.

Waheed, A., Akram, M.U., Khalid, S., Waheed, Z., Khan, M.A., Shaukat, A., 2015. Hybrid features and mediods classification based robust segmentation of blood vessels. J. Med. Syst. 39 (10), 1–14.

Wan, L., Ng, W.K., Dang, X.H., Yu, P.S., Zhang, K., 2009. Density-based clustering of data streams at multiple resolutions. ACM Trans. Knowl. Discov. Data (TKDD) 3 (3), 1–28.

Wang, Z., Chang, S., Zhou, J., Wang, M., Huang, T.S., 2016. Learning a task-specific deep architecture for clustering. In: Proceedings of the 2016 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp. 369–377.

Wang, X.D., Chen, R.C., Yan, F., Zeng, Z.Q., Hong, C.Q., 2019. Fast adaptive K-means subspace clustering for high-dimensional data. IEEE Access 7, 42639–42651. http://dx.doi.org/10.1109/ACCESS.2019.2907043.

Wang, S., Fan, J., Fang, M., Yuan, H., 2014. HGCUDF: Hierarchical grid clustering using data field. Chinese J. Electron. 23 (1), 37–42.

Wang, T.-C., Pham, Y.T., 2020. An application of cluster analysis method to determine Vietnam airlines' ground handling service quality benchmarks. J. Adv. Transp. Hindawi 1–13.

Wang, H., Wang, W., Yang, J., Yu, P.P.S., 2002. Clustering by pattern similarity in large data sets. In: Proceedings of 2002 ACM SIGMOD International Conference Management data – SIGMOD '02, 2, p. 394.

Wang, Y., Yan, H., Sriskandarajah, C., 1996. The weighted sum of split and diameter clustering. J. Classification 13 (2), 231–248.

Wang, J., Zeng, H., Chen, Z., Lu, H., Tao, L., Ma, W.Y., 2003. Recom: reinforcement clustering of multi-type interrelated data objects. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 274–281.

Ward, J.H., 1963. Hierarchical grouping to optimize an objective function. J. Amer. Statist. Assoc. 58, 236–244.

Ware, S., Dhawas, N.A., 2012. Web document clustering using KEA-means algorithm. Int. J. Comput. Technol. Appl. 3 (5), 1720–1725.

Warrenliao, T., 2005. Clustering of time series data—a survey. Pattern Recognit 38 (11), 1857–1874.

Wei, L., Kumar, N., Lolla, V., Keogh, E., 2005. Assumption-free anomaly detection in time series. In: roceedings of the 17th International Conference on Scientific and Statistical Database Management, pp. 237–240.

Weidt, F., Silva, R., 2016. Systematic Literature Review in Computer Science-a Practical Guide. Relatórios Técnicos Do DCC/UFJF, 1.

Wharton, S.W., 1983. A generalized histogram clustering scheme for multidimensional image data. Pattern Recognit. 16 (2), 193–199.

Williams, W.T., Lambert, J.M., 1959. Multivariate methods in plant ecology: I. Association-analysis in plant communities. J. Ecol. 83–101.

Woo, K.-G., Lee, J.-H., 2002. FINDIT: A Fast and Intelligent Subspace Clustering Algorithm using Dimension Voting (Ph.D. thesis). Korea Advanced Institute of Science and Technology, Taejon, Kore.

Wu, J., Zhong, S.-h., Jiang, J., Yang, Y., 2015. A novel clustering method for static video summarization. Multimedia Tools Appl. http://dx.doi.org/10.1007/s11042-016-3569-x.

Xia, Y., Fu, Z., Pan, L., Duan, F., 2018. Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries by order. PLoS ONE 13 (5), e0195457. http://dx.doi.org/10.1371/journal.pone.0195457.

Xie, X.L., Beni, G., 1991. A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 13 (8), 841–847.

Xie, J., Girshick, R., Farhadi, A., 2016. Unsupervised deep embedding for clustering analysis. https://github.com/piiswrong/dec.

Xie, H., Zhang, L., Lim, C.P., Yu, Y., Liu, C., Liu, H., et al., 2019. Improving K-means clustering with enhanced firefly algorithms. Appl. Soft Comput. 84.

Xiong, T., Wang, S., Mayers, A., Monga, E., 2009. A new MCA-based divisive hierarchical algorithm for clustering categorical data. In: Paper Presented At the 2009 Ninth IEEE International Conference on Data Mining.

Xu, D., Tian, Y., 2015. A comprehensive survey of clustering algorithms. Ann. Data. Sci. 2 (2), 165–193.

Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. Neural Netw. IEEE Trans. on 16, 645–678. http://dx.doi.org/10.1109/TNN.2005.845141.

Yager, R., Filev, D., 1994. Approximate clustering via the mountain method. IEEE Trans. Syst. Man Cybern. Part B: Cybern. 24 (8), 1279–1284.

Yaghini, M., Ghazanfari, N., 2020. Tabu-KM: A hybrid clustering algorithm based on tabu search approach. Int. J. Ind. Eng. Prod. Res. (2010), 71–79.

Yan, F., Wang, X.D., Zeng, Z.Q., Hong, C.Q., 2020. Adaptive multi-view subspace clustering for high-dimensional data. Pattern Recognit. Lett. 130, 299–305.

Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M., 2017. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. https://github.com/boyangumn/DCN.

Yang, Y., Ma, Z., Yang, Y., Nie, F., Shen, H.T., 2014. Multitask spectral clustering by exploring intertask correlation. IEEE Trans. Cybern. 45 (5), 1083–1094.

Yang, Y., Wang, H., 2018. Multi-view clustering: A survey. In Big Data Mining and Analytics, Vol. 1, (2). Tsinghua University Press, pp. 83–107. http://dx.doi.org/10.26599/BDMA.2018.9020003.

Yang, J., Wang, W., Wang, H., Yu, P., 2002. δ-Clusters: capturing subspace correlation in a large data set. In: Data Engineering, 2002. Proceedings. 18th International Conference on. pp. 517–528.

Yin, S., Gan, G., Valdez, E.A., Vadiveloo, J., 2021. Applications of clustering with mixed type data in life insurance. Risks 9 (3), 47.

Yu, J., Li, H., Liu, D., 2020. Modified immune evolutionary algorithm for medical data clustering and feature extraction under cloud computing environment. J. Healthcare Eng. 1–11.

Zadeh, L., 1965. Fuzzy sets. Inform. Control 8 (3), 338–353.

Zahn, C.T., 1971. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. C-20 (1), 68–86. http://dx.doi.org/10.1109/T-C.1971.223083.

Zahoor, J., Zafar, K., 2020. Classification of microarray gene expression data using an infiltration tactics optimization (ITO) algorithm. Genes 11 (819), 1–28.

Zare, H., Shooshtari, P., Gupta, A., Brinkman, R.R., 2010. Data reduction for spectral clustering to analyze high throughput flow cytometry data. http://www.biomedcentral.com/1471-2105/11/403.

Zerhari, B., Lahcen, A.A., Mouline, S., 2015. Big data clustering: Algorithms and challenges. In: Conference: LRIT, Unité Associée au CNRST URAC 29. Mohammed V University, Rabat, Morocco.

Zhang, T., Ramakrishnan, R., Linvy, M., 1996. BIRCH: An efficient method for very large databases. In: ACM SIGMOD.

Zhong, C., Miao, D., Wang, R., Zhou, X., 2008. DIVFRP: An automatic divisive hierarchical clustering method based on the furthest reference points. Pattern Recognit. Lett. 29 (16), 2067–2077.

Zhou, Aoying, Cao, Feng, Qian, Weining, Jin, Cheqing, 2008. Tracking clusters in evolving data streams over sliding windows. Knowl. Inf. Syst. 15 (2).

Zhou, L., Wang, D., Guo, L., Wang, L., Jiang, J., Liao, W., 2017. FDS Analysis for multilayer insulation paper with different aging status in traction transformer of high-speed railway. IEEE Trans. Dielectr. Electr. Insul. vol. 24 (5), 3236–3244.

Zhou, Y., Wu, H., Luo, Q., Abdel-Baset, M., 2019. Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowl.-Based Syst. 163, 546–557.

Zolhavarieh, S., Aghabozorgi, S., Teh, Y.W., 2014. A review of subsequence time series clustering. Sci. World J. 312521, 19. http://dx.doi.org/10.1155/2014/312521.